Gesellschaft für Informatik e.V. (GI)

publishes this series in order to make available to a broad public recent findings in informatics (i.e. computer science and information systems), to document conferences that are organized in cooperation with GI and to publish the annual GI Award dissertation.

Broken down into
• seminars
• proceedings
• dissertations
• thematics
current topics are dealt with from the vantage point of research and development, teaching and further training in theory and practice. The Editorial Committee uses an intensive review process in order to ensure high quality contributions.

The volumes are published in German or English.

Information: http://www.gi.de/service/publikationen/lni/

ISSN 1617-5468
ISBN 978-3-88579-677-0

The 11. DFN-Forum Communication Technologies 2018 is taking place in Günzburg, Germany, from June 27th to June 28th.
This volume contains 11 papers selected for presentation at the conference.
To assure scientific quality, the selection was based on a strict and anonymous reviewing process.

P. Müller, B-Neumair, H. Reiser, G. Dreo Rodosek (Hrsg.): 11. DFN-Forum 2018

283

GI-Edition

Lecture Notes in Informatics

Paul Müller, Bernhard Neumair, Helmut Reiser, Gabi Dreo Rodosek (Hrsg.)

11. DFN-Forum Kommunikationstechnologien

27.–28. Juni 2018
Günzburg

Proceedings

Paul Müller, Bernhard Neumair, Helmut Reiser,
Gabi Dreo Rodosek (Hrsg.)

11. DFN-Forum Kommunikationstechnologien

27. – 28. Juni 2018
Günzburg

Gesellschaft für Informatik e.V. (GI)

Lecture Notes in Informatics (LNI) - Proceedings
Series of the Gesellschaft für Informatik (GI)

Volume 283

ISBN 978-3-88579-677-0
ISSN 1617-5468

Volume Editors
Prof. Dr. Paul Müller (pmueller@informatik.uni-kl.de)
Technische Universität Kaiserslautern
Prof. Dr. Bernhard Neumair (bernhard.neumair@kit.edu)
Karlsruher Institut für Technologie (KIT)
PD Dr. Helmut Reiser (reiser@lrz.de)
Leibniz-Rechenzentrum
Prof. Dr. Gabi Dreo Rodosek (Gabi.Dreo@unibw.de)
Universität der Bundeswehr München

printed by Köllen Druck+Verlag GmbH, Bonn

Vorwort

Der **Verein zur Förderung eines Deutschen Forschungsnetzes e. V.** (DFN-Verein) ist eine, von Hochschulen, außeruniversitären Forschungseinrichtungen und forschungsnahen Wirtschaftsunternehmen gegründete Organisation zum Einsatz neuester Kommunikationstechnologien im Bereich seiner Mitglieder. Ziel des Vereins ist die laufende Erneuerung und Optimierung der Netzinfrastruktur sowohl in technischer wie auch ökonomischer Sicht als auch die Entwicklung netznaher Dienste. Aktuelle Beispiele dafür sind die gegenwärtige Plattform des Wissenschaftsnetzes X-WiN sowie die netznahen Dienste DFN-PKI, DFN-AAI und föderierte Cloud-Dienste zur Unterstützung von Lehre und Forschung in den Mitgliedseinrichtungen. Um diese Technologien und Dienste einerseits selbst mit zu gestalten und andererseits frühzeitig die eigenen Entwicklungen mit denen anderer Wissenschaftler abzugleichen, veranstaltet der DFN-Verein seit vielen Jahren das DFN-Forum Kommunikationstechnologien. Mit den Zentren für Kommunikation und Informationsverarbeitung in Forschung und Lehre e.V. (ZKI) und der Gesellschaft für Informatik e.V. (GI) gibt es in diesem Bereich eine langjährige und fruchtbare Zusammenarbeit.

Das DFN-Forum Kommunikationstechnologien „Verteilte Systeme im Wissenschaftsbereich“ im Jahr 2018 ist bereits die 11. Veranstaltung in dieser Reihe. Es setzt die Tradition der zehn sehr erfolgreichen Vorgänger in Kaiserslautern, München, Konstanz, Bonn, Regensburg, Erlangen, Fulda, Lübeck, Rostock und Berlin fort. Das diesjährige Forum ist eine länderübergreifende Veranstaltung, die vom DFN-Verein und der Universität Ulm gemeinsam mit dem ZKI e.V. und der GI veranstaltet wird. Die Veranstaltung, von einer baden-württembergischen Universität organisiert, wird in Bayern, zum ersten Mal sogar in einem Schloss, stattfinden. Am 27. und 28. Juni 2018 wird das Wissenschaftszentrum Schloss Reisensburg, in der Nähe von Günzburg, Veranstaltungsort des Forums sein. Wie seine Vorgänger soll es eine Plattform zur Darstellung und Diskussion neuer Forschungs- und Entwicklungsergebnisse aus dem Bereich TK/IT darstellen. Das Forum dient dem Erfahrungsaustausch zwischen Wissenschaftlern und Praktikern aus Hochschulen, Großforschungseinrichtungen und Industrie.

Aus den eingereichten Beiträgen konnte ein hochwertiges und aktuelles Programm zusammengestellt werden, das sich neben Fragen von IT-Sicherheit und IT Service Management auch mit Infrastrukturen für Forschung, Lehre sowie entsprechender Testbeds befasst. Grundlegende Fragen und Entwicklungen der Netztechnologien sind ureigene Themen im Kontext von Forschungsnetzen und werden im Rahmen einer eigenen Sitzung auf dem Forum behandelt. Ein Tutorial zu Blockchain-Technologie und Bitcoins, sowie eingeladene Beiträge zum 100 Gbit Netz in Baden-Württemberg sowie das neue Forschungsinstitut CODE an der Universität der Bundeswehr München, runden das Programm ab.

Wir möchten uns bei den Autoren für alle eingereichten Beiträge, beim Programmkomitee für die Auswahl der Beiträge und die Zusammenstellung des Programms, bei den Mitarbeiterinnen und Mitarbeitern der Geschäftsstelle des DFN-Vereins für die Organisation und beim Gastgeber für die Unterstützung des Forums sowie die Gastfreundschaft bedanken.

Unser besonderer Dank gilt Prof. Dr. Paul Müller für die langjährige und vertrauensvolle Zusammenarbeit im DFN-Forum. Er hat das Forum von Beginn an begleitet und aktiv mitgestaltet. Sein großes Engagement und seine Erfahrung haben das DFN-Forum positiv bereichert. Wir wünschen ihm einen erfüllten und unbeschwerten Ruhestand und alles Gute für die Zukunft.

Allen Teilnehmern wünschen wir für die Veranstaltung interessante Vorträge und fruchtbare Diskussionen.

Reisensburg, Mai 2018

Gabi Dreo Rodosek
Paul Müller
Bernhard Neumair
Helmut Reiser

Programmkomitee

Rainer Bockholt, Universität Bonn

Alexander Clemm, Huawei USA

Gabriele Dobler, Landesamt für die Sicherheit in der Informationstechnik, Nürnberg

Gabi Dreo Rodosek (Co-Chair), Universität der Bundeswehr München

Thomas Eickermann, Forschungszentrum Jülich

Alfred Geiger, T-Systems SfR

Andreas Hanemann, Fachhochschule Lübeck

Ulrich Lang, Universität zu Köln

Paul Müller (Co-Chair), Technische Universität Kaiserslautern

Bernhard Neumair (Co-Chair), KIT

Christa Radloff, Universität Rostock

Helmut Reiser (Co-Chair), LRZ München

Sebastian Rieger, Hochschule Fulda

Harald Roelle, Siemens AG

Uwe Schwiegelshohn, TU Dortmund

Marcel Waldvogel, Universität Konstanz

René Wies, BMW Group

Stefan Wesner, Universität Ulm

Martin Wimmer, Deutsches Zentrum für Neurodegenerative Erkrankungen e.V. (DZNE)

Inhaltsverzeichnis

Tutorial

Sicherheit

IT-Management, Lehre und Testbeds

Netztechnologien

The Bitcoin Universe: An Architectural Overview of the Bitcoin Blockchain

Paul Mueller[1,2], Sonja Bergsträßer[2], Amr Rizk[2] and Ralf Steinmetz[2]

Abstract: On January 2009, the emergence of Bitcoin surprised the world with a new idea involving decentralized secure money transfers outside the ecosystem of FIAT currencies. The concepts behind the Bitcoin architecture the blockchain can, however, be extended to a much wider range of economic assets than just digital currencies. In general, a blockchain is a distributed, verifiable database, which operates through a confluence of public-key cryptography, the concept of *proof of work* and P2P-systems.

In the following article, a detailed overview of the concepts underlying the Bitcoin architecture is given. The ecosystem as a whole is discussed, starting with some historical aspects. We consider the cryptographic background as given here, as this article discusses mainly the basic concepts for key generation. Here, we focus on the concept of transactions in the scope of Bitcoin which we break down into single attributes such as the locking and unlocking scripts. Both, the mining process and the consensus mechanism are examined. Furthermore, the drawbacks of the whole system and the proposed remedies to these via Bitcoin Improvement Proposals (BIP) are outlined. Finally, we address several applications based on the blockchain as well as the question of anonymity.

Keywords: Bitcoin, blockchain, proof-of-work (PoW), Bitcoin keys, Transactions scripts, Mining, Consensus, Bitcoin drawbacks, Bitcoin applications

1 Introduction

The "big bang" of the Bitcoin blockchain took place on January 3rd, 2009, when the first Bitcoin block, the genesis block, was established at 18:15:05 GMT. The genesis block is the only block in the blockchain which is hard-coded within the source code of Bitcoin, rather than the result of the mining process. However, the story of Bitcoin started in 2008, when the domain name "bitcoin.org" was registered anonymously. The basic ideas of the blockchain were then published[3] by "Satoshi Nakamoto" on Friday, October 31st,[4] 2008 at 18:10:00 UTC under the title "Bitcoin: A Peer-to-Peer Electronic Cash System" [SN08]. The author(s) of this document is/are still unknown.

The complexity of the Bitcoin ecosystem comes from it aims, i.e., that is that anyone should be able to write to the Bitcoin blockchain, and that there should be no centralized

[1] University Kaiserslautern, Integrated Communication Systems Lab (ICSY), Paul Ehrlichstraße 34, 67663 Kaiserslautern, pmueller@informatik.uni-kl.de
[2] TU Darmstadt, KOM Multimedia Communications Lab, Rundeturmstraße 10, 64283 Darmstadt, {Vorname.Nachname}@KOM.TU-Darmstadt.de
[3] http://article.gmane.org/gmane.comp.encryption.general/12588/
[4] Halloween

control. The Bitcoin ecosystem can be viewed as a network of replicated databases, where each database contains the same list of previous Bitcoin transactions. Full nodes (nodes who runs the full stack of the Bitcoin protocol) of the network are called “miners”, and these propagate “transaction data” (payments) and “block data” (additions to the ledger). Each miner independently checks the transaction and block data passed to it. There are rules in place (the Bitcoin protocol) to make the network operates as intended. The complexity of the Bitcoin architecture arises from its aims, which are to be decentralized, that is, to have no single point of control, and to be highly secure and anonymous. This has influenced how Bitcoin has developed. All blockchain ecosystems need not have the same mechanisms, especially if participants can be identified and trusted to behave (e.g. in a private blockchain).

The outline of this paper is as follows: Firstly, several historical remarks are made; these are followed by a description of the architectural design principles involved. This description starts with Bitcoin keys, followed by a detailed description of transactions and the mining process, and finishes with a discussion of the consensus approach used in Bitcoin and the vulnerability of this architecture. Next, the drawbacks of Bitcoin architecture are described, followed by countermeasures to these drawbacks as put forward in Bitcoin Improvement Proposals (BIP). The last chapter concerns Bitcoin applications, where some concrete applications are introduced. In the conclusion of the paper, we discuss the question of anonymity.

2 The Architecture of Bitcoin

With respect to the architecture (see Fig. 1) of the Bitcoin blockchain, several important design aspects must be taken into account:

1. The Bitcoin application itself
2. The role of nodes constituting the overall blockchain network, and the node discovery process
3. Transactions, which make up the blocks running in the nodes
4. The security implementation that generates the blocks
5. The process of adding new blocks to the chain.

The blockchain itself runs on a network of distributed servers. The core application is a transaction database modeled as a secure ledger. This is shared by all nodes (servers) that run the full stack of the Bitcoin protocol. It is thus a decentralized transaction system acting as a highly transparent ledger. Any full node running the blockchain protocol runs the entire blockchain locally.

After installing the full stack of the software, the blockchain client syncs up with the other nodes in the network, in a peer-to-peer fashion [SW05]. Hence, that node maintains all Bitcoin transactions (or any other application running on the blockchain). The integrity and chronological order of transactions (and the addresses owning the currency) are enforced by cryptographic rules.

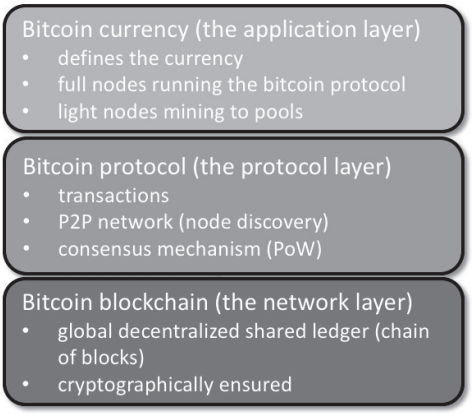


Fig. 1: The Bitcoin Architecture

The nodes in the overall network use a peer-to-peer IP network to process and verify transactions. Nodes that have the same blocks in their individual databases are considered to be in consensus [DW13].

2.1 The Design of the Bitcoin Blockchain

We now take a closer look at the Bitcoin blockchain. As mentioned above, the first block of the Bitcoin blockchain was not a result of the Bitcoin consensus mechanism (mining) but was hard-coded into the source code. It is a special case, in the sense that it does not reference a previous block, and for Bitcoin and almost all of its spinoffs, it produces an unspendable subsidy (for a detailed description of the genesis block, see https://en.bitcoin.it/wiki/Genesis_block).

At least one parameter of the Genesis block is worth a deeper look. The "coinbase parameter" (coded in hex) contains, along with normal data, the following text:

"The Times 03/Jan/2009 Chancellor on brink of second bailout for banks"

This may be intended as proof that the block was created on or after January 3rd, 2009, as mentioned above; it could also form an argument for a new currency, due to the instability caused by traditional banking.

Another architectural design decision was the limited total number of Bitcoins, resulting from the speed of mining (evaluating) a new block and the reward that a miner can earn from the mining process. On average, a new block is mined every 10 minutes (regardless of the technology used, as discussed later), and the mining reward at the beginning of the system was 50 Bitcoins per block. The block reward is halved every four years (or every 210,000 blocks on average). A simple calculation shows that there will be a total of 21 million Bitcoins available as reward for miners.

Another design decision was the block size. Currently, the block size is restricted to 1MB (on average), and one block can therefore cover around 4000 transactions with an average size of 250 bytes. This results in an overall rate of about seven transactions per seconds (tps). Compared to PayPal (around 200 tps) and VISA (4,000–40,000 tps), this is a fairly low rate. An examination of https://blockchain.info/ gives information on how many transactions are currently included in each block, although there are some blocks with only one transaction, meaning that the miner included no transactions except for their own reward transaction[5].

2.2 The Steps of a Bitcoin Transaction

Transactions can be broadcast by any node in the system at any time. The decision on which transactions of those broadcasted to be included in a new block is dependent on the node (the miner) running the ***proof-of-work*** **(PoW)** algorithm, since the miners are responsible for picking a transaction from the so-called *mem-pool* where all validated transactions[6] are stored, grouping them and including them in the block. The selection of transactions by the miner depends on the transaction fee (the standard fee is 1.000 Satoshi = 10 μBTC = 0.01 mBTC = 0.0001 BTC per kB), which forms a reward for the miner's efforts in addition to the coinbase reward. The priority of picking up a transaction depends on the miner; in general, miners prefer larger fees and smaller transactions, and often prioritize in this way. Transaction portions that is not spent towards recipient of back to the sender are included as a fee. Fees are paid to miners and can be used to increase the speed of transaction confirmation by incentivizing miners to prioritize the transaction(s).

To initiate a transaction, a user must generate the necessary Bitcoin keys (see Fig. 2), starting with a random 256-bit private key (see Step 1). This **private key** is needed to sign a transaction and thus transfer (spend) Bitcoins. The elliptic curve DSA algorithm is then used to generate a 512-bit **public key** from the private key (see Step 2). This public key is used to **verify** the signature for a transaction, although the public key is not revealed until a transaction is signed. Since the 512-bit public key is inconveniently large, it is hashed down to 160 bits using the SHA-256 and RIPEMD hash algorithms (see Step 3).

[5] https://blockchain.info/block-height/315076

[6] validated transaction: all transactions (payments) which follows the Bitcoin protocol rules

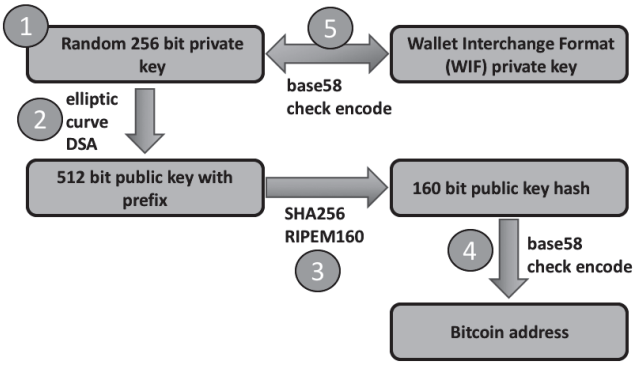


Fig. 2: Bitcoin keys and their relationships[7]

The key is then encoded in ASCII using Bitcoin's custom Base58Check encoding (see Step 4). The resulting address is the **Bitcoin address**, which is published in order for a user to receive Bitcoins. Note that neither the public key nor the private key can be determined from the Bitcoin address. If the private key is lost (for instance, by throwing a hard drive[8] away), the Bitcoins are lost forever.

When all keys have been generated, a **transaction** can be carried out. A transaction enables a transfer of Bitcoins and is broadcast to the network[9] nodes. Transactions are the main building blocks of the Bitcoin system. The Bitcoin architecture is designed to make sure that transactions can be transparently added to the global ledger of transactions. Here, there is even no need to trust the nodes used to broadcast the transaction.

Transactions are data structures that encode a transfer of value between participants in the Bitcoin system. Each transaction is a public entry in the Bitcoin blockchain, and a transaction chain is formed (see Fig. 3), meaning that all users can see every Bitcoin transaction from the genesis block onwards.

[7] Adopted from http://www.righto.com/2014/02/bitcoins-hard-way-using-raw-bitcoin.html
[8] https://www.theguardian.com/technology/2013/nov/27/hard-drive-bitcoin-landfill-site
[9] There are several scripts available in Bitcoin to handle the value transfer from the source to the destination. These scripts can handle very simple to relatively complex transactions, depending upon requirements.

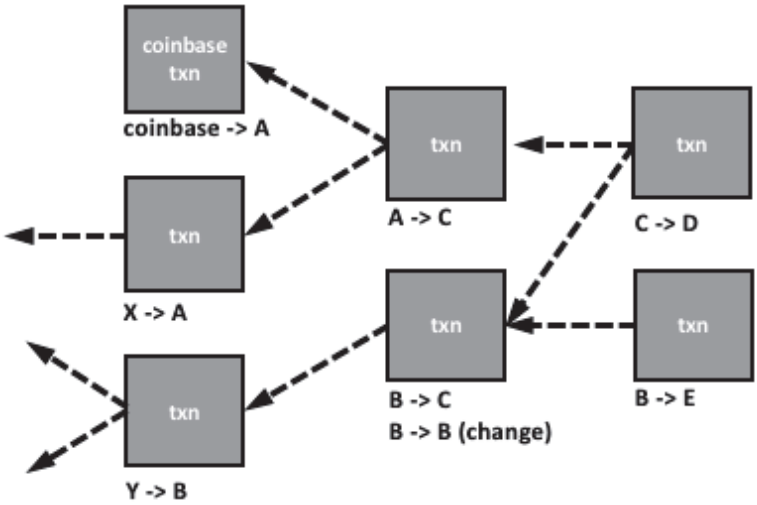


Fig. 3: The Bitcoin transaction chain[10]

A transaction usually references previous transaction outputs as new transaction inputs and uses all input Bitcoin values. Thus, one key element of a Bitcoin transaction is an unspent transaction output (UTXO). UTXOs are indivisible pieces of Bitcoin currency locked to a specific owner that are recorded in the blockchain. Although a UTXO can have any arbitrary value, it is indivisible once created, just like a coin that cannot be cut in pieces. When a user receives a Bitcoin value, the amount is recorded within the blockchain as a UTXO. Thus, one Bitcoin may be scattered in the form of UTXOs across hundreds of transactions and hundreds of blocks. There are no accounts or balances in Bitcoin; there are only dispersed UTXOs, locked to specific owners in the transaction chain. This means that each input used must be entirely spent in a transaction. If an address received 12.5 Bitcoins and wants to spend only 12.0 Bitcoins, the transaction must spend all 12.5. Hence, one uses a second output (the change address) for ***the difference***, which returns the 0.5 leftover Bitcoins back to the sender (see Fig.4).

Transactions in the blockchain are not encrypted, and it is therefore possible to browse and view every transaction ever made, each of which is stored in a block related to the Bitcoin address rather than the name of the user. Once transactions receive sufficient confirmations, they can be considered irreversible. All transactions are visible (the transaction chain which gives the history of ownership) in the blockchain (which gives the

[10] Adopted from http://www.imponderablethings.com/2013/07/how-bitcoin-works-under-hood.html

transaction ordering in time to avoid double spending) and can be viewed with a hex editor. The transaction itself has the general format shown in Fig. 5.[11]

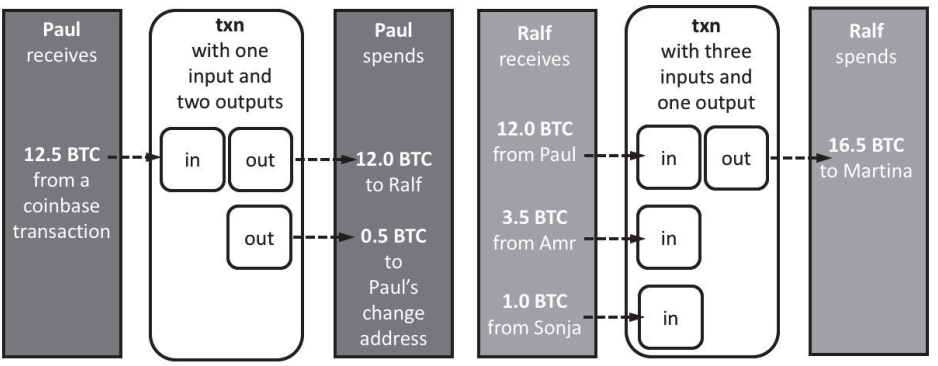


Fig. 4: Input and outputs of a transaction

In general, a Bitcoin transaction is composed of three parts:

- **An input:** This is a record of the Bitcoin address from which the Bitcoins to be spent were initially received.
- **An amount:** This is the specific amount of Bitcoin to be sent.
- **An output:** This is the receiver's public key, also known as a 'Bitcoin address.'

The transaction **input** (or list of inputs) is a pointer to a UTXO using a reference to the transaction hash and sequence number of the record in the blockchain. To spend a UTXO, a transaction input also includes an **unlocking script** that satisfies the spending conditions set by the UTXO. The unlocking script is usually a signature proving the ownership of the Bitcoin address in the locking script. The unlocking script is called *scriptSig*, because it usually contains a digital signature.

A transaction **output** (or list of outputs) comprises, first, an amount of Bitcoins, denominated in satoshis (the smallest Bitcoin unit), and secondly, a **locking script** to "lock" this amount by specifying the conditions that must be met to spend the output. The locking script is called *scriptPubKey*, because it usually contains a public key or Bitcoin address.

[11] From https://en.bitcoin.it/wiki/Transaction#general_format_.28inside_a_block.29_of_each_input_of_a_transaction_-_Txin

field	description	size
Version no	currently 2	4 bytes
In-counter	positive integer VI = VarInt	1 – 9 bytes
List of inputs	the first input of the first transaction is also called "coinbase"	<in-counter> many inputs
Out-counter	positive integer VI = VarInt	1 – 9 bytes
List of outputs	the outputs of the first transaction spend the mined bitcoins for the block	<out-counter> many outputs
Lock_time	if non-zero and sequence numbers are < 0xFFFFFFFF: block height or timestamp when transaction is final	4 bytes

Fig. 5: General transaction format (from: https://en.bitcoin.it/wiki/Transaction)

Breaking down a concrete transaction script (with only a single input and output) and deserializing it, we obtain the following structure, where values are displayed in hexadecimal.

```
01  {"hash":"8ac3455…",
02  "ver":1,
03  "vin_sz":1,
04  "vout_sz":1,
05  "lock_time":0,
06  "size":224,
07  "in":[
08    {"prev_out":
09      {"hash":"aee433…",
10      "n":0},
11    "scriptSig":<sig> <pubKey>}],
12  "out":[
13    {"value":"0.50000000",
14     "scriptPubKey":"OP_DUP OP_HASH160 <pubKeyHash?>
                      OP_EQUALVERIFY OP_CHECKSIG"}]}
```

Algorithm 1: Inside a simple transaction (one input and one output)

In the following, algorithm 1 is explained line by line. In Line 1, we see the remainder of a transaction, `8ac3455...`. This is used as an identifier for the transaction, followed by the version number of the Bitcoin protocol in Line 2 and the number of inputs and outputs (one of each in this case) in Lines 3 and 4. The `lock_time` parameter in Line 5 controls when a transaction should be finalized. For most Bitcoin transactions carried out today, the `lock_time` is set to 0, which means the transaction is finalized immediately. The size of the transaction in bytes is described in Line 6.

Lines 7 to 11 define the transaction input, which is taken from the output of an earlier transaction, and uses the hash of this earlier transaction in Line 9. The parameter n in Line 10 denotes that this is the first output from that transaction. The *scriptSig* parameter in Line 11 contains the signature of the sender, followed by a space, and then the corresponding public key (the unlocking script).

Lines 12 to 14 give the output of the transaction: Line 13 gives the value of the output, and Line 14 contains the *scriptPubKey* (the unlocking script) value with the Bitcoin address (<pubKeyHash?>) of the intended recipient of the funds. The other parameters used here will be described later.

To verify a transaction, the Bitcoin validation engine relies on two types of scripts: a *locking script* and an *unlocking script*, as shown in Fig. 7.

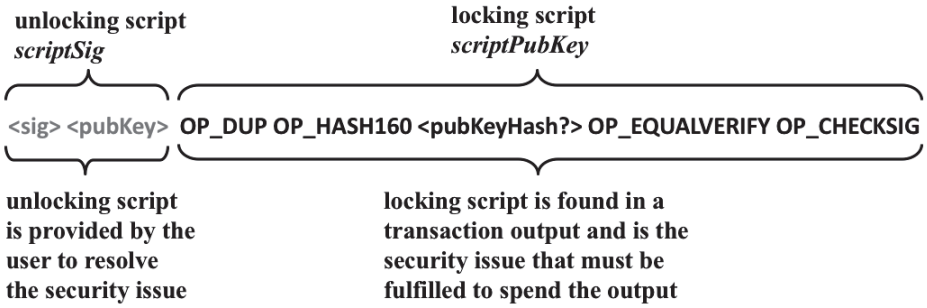


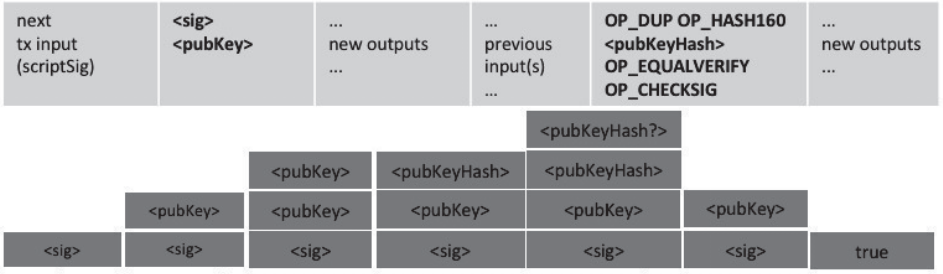
Fig. 6: Bitcoin locking and unlocking scripts (from: Mastering Bitcoin [AA17])

Here, only the standard 'pay to public key hash' (P2PKH) transaction will be discussed. P2PKH is the most commonly used transaction type, and is used to send transactions to Bitcoin addresses. Bitcoin uses a simple stack-based[12] language called a *script* to describe how Bitcoins can be spent and transferred. It is intentionally not Turing complete, and has no loops to avoid a condition where a script runs forever. This scripting language uses a reverse Polish notation in which every operand is followed by its operators. It is evaluated from left to right, using a 'last in first out' (LIFO) stack. A script uses various opcodes (operational codes) to define its operation.

Each node examines a transaction as it arrives, and then runs a series of checks to verify it. These checks[13] are defined in the protocol rules. Every Bitcoin node validates

[12] Note: The stacks hold byte vectors. When used as numbers, byte vectors are interpreted as little-endian variable-length integers, with the most significant bit determining the sign of the integer. Byte vectors are interpreted as Booleans, where False is represented by any representation of zero and True is represented by any representation of non-zero.
[13] https://en.bitcoin.it/wiki/Protocol_rules

transactions by executing the locking and unlocking scripts simultaneously[14]. For each input in the transaction, the validation software first retrieves the UTXO referenced by the input. This UTXO contains the *locking script* that defines the conditions required to spend it. The validation software then takes the *unlocking script* contained in the input attempting to spend this UTXO, and executes these two scripts. First, the UTXO is unlocked, and then it is spent. *scriptSig* is provided by the user who wishes to unlock the transaction, while *scriptPubKey* is part of the transaction output and specifies the conditions that need to be fulfilled in order to spend the output.



	stack	script	description
1	empty	<sig> <pubKey> OP_DUP OP_HASH160 <pubKeyHash> OP_EQUALVERIFY OP_CHECKSIG	scriptSig and scriptPubKey are combined.
2	<sig><pubKey>	OP_DUP OP_HASH160 <pubKeyHash> OP_EQUALVERIFY OP_CHECKSIG	constants are added to the stack.
3	<sig><pubKey> <pubKey>	OP_HASH160 <pubKeyHash> OP_EQUALVERIFY OP_CHECKSIG	top stack item is duplicated.
4	<sig><pubKey> <pubKeyHash>	<pubKeyHash> OP_EQUALVERIFY OP_CHECKSIG	top stack item is hashed
5	<sig><pubKey> <pubKeyHash> <pubKeyHash?>	OP_EQUALVERIFY OP_CHECKSIG	constant added
6	<sig><pubKey>	OP_CHECKSIG	equality is checked between the top two stack items
7	true	empty	signature is checked for top two stack items

Fig. 7: Bitcoin stack operation

A script is an instruction list paired with each transaction to describe how to gain access to the transferred Bitcoins. Fig. 8 shows a step-by-step execution of the combined script, which will determine whether or not this is a valid transaction. The execution starts (from left to right) by pushing the value <sig> onto the stack. Next, the value <pubKey> is

[14] In the original Bitcoin client, the unlocking and locking scripts were concatenated and executed in sequence. For security reasons, this was changed in 2010.

pushed onto the stack, and this is duplicated in the next step with the operator DUP. The operator HASH160 hashes the top item of the stack using RIPEMET160(SHA256(pubKey)), and pushes the resulting value onto the stack.

The value <pubKeyHash?> from the script is pushed on top of the value <pubKeyHash> previously calculated from the HASH160 of the <pubKey>. The EQUALVERIFY operator compares the <pubKeyHash?> from the script with the <pubKeyHash?> calculated from the user's <pubKey>. If they match, both are removed from the stack and execution is continued. Finally, the CHECKSIG operator checks that the signature <sig> matches the public key <pubKey> and pushes TRUE onto the stack if true. After successful validation, the transaction is copied into the memory pool (mem-pool) of the node and waits to be picked up and included into a block.

Every full node maintains a temporary list of unconfirmed transactions that is called the *mem-pool*. Nodes use this pool to keep track of transactions that are known to the network but have not yet been included in the blockchain. As Transactions that become part of a block and are added to the blockchain are considered "confirmed", allowing the new owners to spend the Bitcoin they received in these transactions. In the following, the process by which a transaction becomes part of a block is described.

2.3 The Bitcoin Blockchain: The Mining Process

The blockchain is a public ledger providing a time stamped, ordered, and immutable list of all transactions on the Bitcoin network. Due to the design decision in the Bitcoin network whereby each block cannot exceed a limit of 1MB, the average number of transactions per second is around seven, i.e., a fairly low number. Since transactions can be launched any time, the number of transactions waiting in the mem-pool[15] tends to increase, and reached about 50 million in January 2018.

After validation, the transaction must be included into a block. To do this, the miner (the node) picks transactions from the mem-pool, recursively hashing pairs of transaction hash values until there is only one remaining; this is called the Merkle root [RM80]. The cryptographic hash algorithm used in Bitcoin's Merkle trees is SHA256, applied twice, also known as double-SHA256.

We now take a closer look at a block respectively the block chain (see Fig. 9). Consider a block as a data structure that aggregates transactions. The block consists of a header with some metadata, mainly followed by the list of transactions. The block header has a size of 80 bytes, while the average transaction is at least 250 bytes and a block can contain up to 4000 transactions.

[15] Mem-pool size: https://blockchain.info/charts/mempool-size?

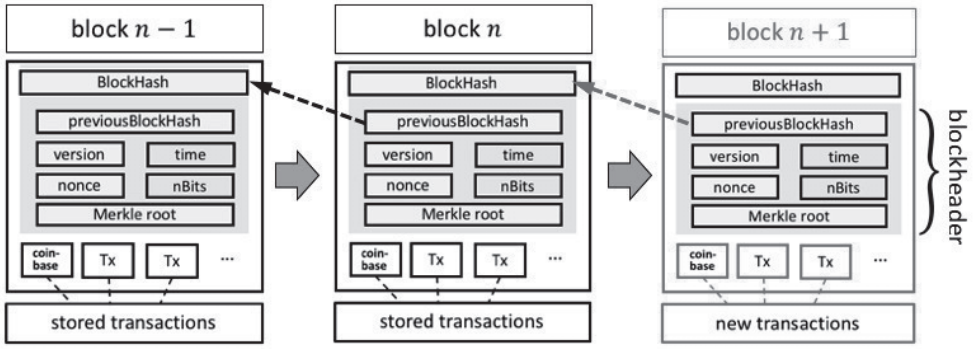


Fig. 8: The blockchain

The block header consists of a reference to a previous block hash; this connects the block to the previous block in the blockchain. The second set of metadata, containing the target, timestamp and nonce, relate to the mining competition. The third part of the metadata is the Merkle tree root, as described above.

When a miner (a node) has put these data together, it can start the process of linking the new block to the blockchain. As an incentive to miners to spend computing power in the mining process, they receive two types of rewards for mining. Firstly, they earn new coins, which are created with each new block in the so called *coinbase* transaction automatically. That grants the mining reward to the miner. Secondly, they receive transaction fees from the remaining transactions included in the block which are proportional to the size (1000 satoshis per kB).

To earn these rewards, all miners compete to solve a difficult mathematical puzzle based on a cryptographic hash algorithm. The solution to this problem, i.e., the proof of work (PoW), is copied into the new block. The PoW is seen as proof that a miner spent significant computing effort. The competition to solve the PoW algorithm to earn rewards and the right to add transactions to the blockchain is the basis of Bitcoin's security model.

To run the PoW, the miner first has to hash the block header, including the block hash of the previous block, and compare it with a predefined target which is stored in the nBits parameter of the header. Also the target is formally defined as $target = \mathbf{2^{224}/d}$ (where $\boldsymbol{d}$ denotes the difficulty). In the concrete implementation it can be retrieved from the *hex* representation of the block header (80 bytes in total) using the 73-76 byte. Because this number, however, is in little-endian the bytes have to be reversed. Now the target can be calculated from a compact scientific notation ($target = c*2^{(8\ (e-3))}$) where the first byte denotes the exponent e and the next 3 bytes the coefficient c. The proof of work is formally defined as:

$$\mathbf{PoW} = F_d(c|x) \rightarrow \mathbf{SHA256(SHA256}(h|x) \leqq \mathbf{2^{224}/d}$$

where $\boldsymbol{h}$ can be seen as a *challenge*, $\boldsymbol{x}$ the *nonce* and $\boldsymbol{d}$ the actual *difficulty* [RW16]. If $\boldsymbol{F_d(h|x)}$ is not smaller than the current target, the miner will modify the nonce (usually just incrementing it by one) and calculate the block hash again. At the current level of difficulty in the Bitcoin network, miners have to try quadrillions of times before finding a nonce that results in a low enough block header hash. The length of time it takes to mine a block can be controlled with the difficulty,[16] $\boldsymbol{d}$. In order to keep the block generation time to 10 minutes on average regardless of the technology (increasing compute power of miners) used, the difficulty of mining must be adjusted. In fact, difficulty is a dynamic parameter that is periodically (every 2016 blocks or every 14 days) adjusted to meet a 10-minute block target. Adjustment of the difficulty occurs automatically and independently on every full node after every 2016 blocks, and all nodes retarget the proof of work difficulty. The equation for retargeting difficulty compares the time to find the last 2,016 blocks to 20,160 minutes, i.e., the time needed based on the 10-minute average block generation time. This can be put in simple terms as follows: if the network is finding blocks faster than every 10 minutes, the difficulty increases; if block creation time is slower than expected, the difficulty decreases.

The last step in Bitcoin's consensus mechanism is the independent validation of each new block by every node in the network. As the newly solved block is propagated across the network, each node performs a series of tests to validate it before propagating it to its peers. This ensures that only valid blocks are propagated on the network. This independent validation also ensures that miners who act honestly have their blocks incorporated in the blockchain, thus earning the reward.

2.4 The Bitcoin Consensus

While mining is primarily incentivized by the generation of rewards, the underlying purpose of mining is not the reward or the generation of new coins but the achievement of a decentralized consensus in a trustless network. Mining enables a network-wide consensus without a central authority.

The first practical implementation of a distributed consensus in a trustless network was realized by Bitcoin, which uses public key cryptography with PoW based on hashcash[17] [AB02]. The key innovation here is the idea of an ordered list of blocks composed of transactions and cryptographically secured by the PoW mechanism. Bitcoin's decentralized consensus arises from the interplay of four processes occurring independently on nodes across the network:

[16] The block header also contains the difficulty target in a notation called "difficulty bits" or just "nBits." This is expressed in a coefficient/exponent format, with the first two hexadecimal digits representing the exponent and the next six hex digits the coefficient.

[17] http://www.hashcash.org/

- Independent verification of each transaction by every full node, based on a comprehensive list of criteria (https://en.bitcoin.it/wiki/Protocol_rules).
- Independent aggregation of these transactions into new blocks by mining nodes, coupled with computational efforts using a proof of work algorithm.
- Independent verification of the new blocks by every node, and assembly into a chain.
- Independent selection by every node of the chain with the most cumulative computational effort, demonstrated through proof of work.

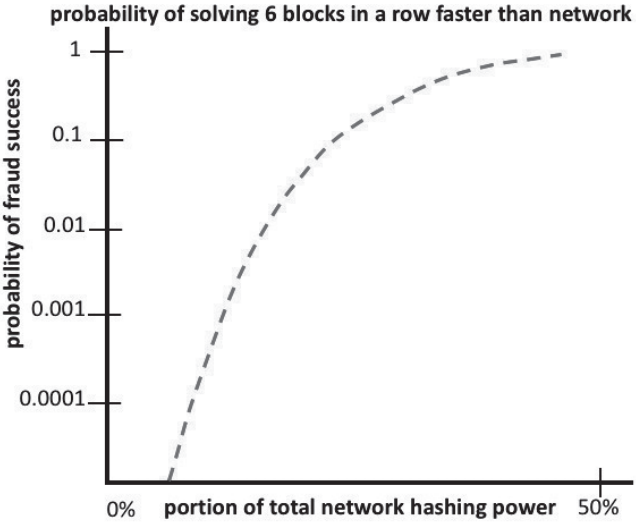


Fig. 9: The 51% attack (calculated from M. Rosenfeld [Ro12])

Because of the distributed nature of Bitcoin, two nodes may announce a valid block simultaneously, meaning that there exist two blockchains containing different transactions. This situation depends on the block creation time relative to the block propagation time, and is addressed by the Bitcoin network accepting only the longest chain. In this case, the shorter chain will be considered orphaned; this is a point at which the Bitcoin network is vulnerable.

This vulnerability is known as the 51% attack (see Fig. 9). In this scenario, a miner or group of miners (mining pools) that control a majority (51%) of the total network's hashing power can attack the Bitcoin blockchain. In this type of attacks, the attackers can leverage/cause forks in the blockchain for their own benefit. For example, one could double-spend transactions or "prohibit" transactions from executing. A double-spend attack is one where the attacker deliberately causes forks at a previous position in the

blockchain. With sufficient power, an attacker can invalidate six or more blocks in a row, causing transactions that were considered immutable to be invalidated.

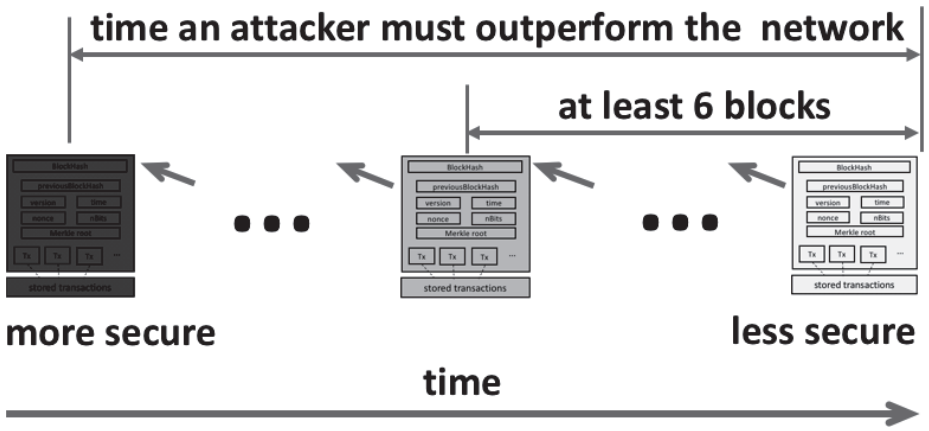


Fig. 10: Security timeline[18]

As explained above, the chance of a single miner solving a new block before the rest of the network, which takes 10 minutes on average, is very low. Even with substantial computing power, the older a transaction/block becomes, the harder it would be for an attacker to change it (Fig. 11).

3 The Drawbacks of Bitcoin

Although the Bitcoin network is the most well-known example of a cryptocurrency, there is currently a discussion about its technical drawbacks.

3.1 Defining the Drawbacks

Firstly, the fairly **small transaction time** of 7 tps, which is a result of the defined block size of 1MB and the average block creation time of 10 minutes, is under discussion. This limitation, together with the high popularity of Bitcoin, has resulted in a dramatic increase in the mem-pool (more than 50 million transactions are currently waiting in the mem-pool[19]) and therefore in a massive delay in transactions.

Another drawback concerns **scalability**. The size of the Bitcoin blockchain reached approximately 154GB in February 2018, and is growing at a rate of 6MB per hour (52GB

[18] Adopted from: http://www.imponderablethings.com/2013/07/how-bitcoin-works-under-hood.html
[19] https://blockchain.info/charts/mempool-size?

per year on average). Although storage capacity is not a real issue today, it is likely to become an issue for small devices, possibly in the IoT arena.

Bitcoin **transaction fees** are also an issue because they depend on the actual Bitcoin market value. This depends on a variety of factors, and some wallets allow users to manually set transaction fees, but the standard fee is about 0.0001BTC/kB. At the current market value of about $8,700 per BTC, the standard transaction fee is about $8 with a very high volatility ($52 as of December 23rd, 2017). For small transaction amounts, this fee is therefore prohibitively high (for more information, see https://bitinfocharts.com/[20]).

At the outset of the Bitcoin network, it was possible to run the proof of work algorithm on the CPUs of nearly all computers. However, with the emergence of GPU and ASIC hardware, it has become increasingly difficult for single mining nodes to stay in the race. At this time, mining pools or groups of cooperating miners who agree to share block rewards in proportion to their contributed mining hash power have come into play. While these are desirable for the average miner, they unfortunately concentrate power within the mining pools. Today there are about 20 major mining pools. Relative to the hash power controlled by a pool and its location, it can be concluded that Chinese pools control approximately 81% of the network hash rate. This de facto **centralization** contradicts Satoshi Nakamoto's original idea of a real decentralized network of cryptocurrency, and poses a threat to the core concepts underlying Bitcoin.

Last but not least, the current **power wastage** of the Bitcoin network has been widely discussed in the media, since all transactions and the mining of blocks (the PoW effort) are done independently by every miner in the world. When a miner has successfully mined a block, it is propagated to all the others, resulting in the current mining process being terminated and all other miners losing the race for rewards, thus wasting all the invested energy. The current energy consumption due to Bitcoin (as of February 2nd, 2018)[21] is about 46.8TWh; this is the equivalent of the energy consumption of a small country such as Denmark.[22] This should, however, be viewed in comparison with the power consumption due to Google searches, Facebook usage and the data centers of the financial industry, which is also known to be extremely high.

3.2 Countermeasures

These issues reviewed above are well known to the Bitcoin community, which is working hard to overcome these shortcomings. The main approaches are described in the Bitcoin Improvement Proposals (BIP) which can be found on Github.[23] Numerous scaling solutions have been proposed for the above drawbacks.

[20] https://bitinfocharts.com/
[21] https://digiconomist.net/bitcoin-energy-consumption
[22] https://en.wikipedia.org/wiki/List_of_countries_by_electricity_consumption
[23] https://github.com/bitcoin/bips/blob/master/README.mediawiki.

The first intuitive approach for increasing the tps is to **expand the block size**. The current block size of 1MB was set in 2010 for security reasons, to prevent miners from creating large spam blocks. Since the transaction volume has increased with the widespread usage of Bitcoin, an increase in the limit of 1MB became the subject of vociferous debate in 2015. This debate is still ongoing, and the main focus is on whether such an expansion should support the old version of 1MB (soft fork) or not (hard fork).

Another option that has been discussed is to push the myriad tiny transactions from gambling or crowd-working sites "off-chain". This approach is known as the **Lightning Network**[24], whereby transactions are sent over a network of micropayment channels. This would allow two users to carry out their transactions in a private room, and to then add their data back on the blockchain at an agreed time. However, even this would require a soft fork of the protocol to get started. The most prominent contender to apply a scaling solution based on side chains such as the Lightning Network is Blockstream,[25] which offers paid side chain services.

Miners are compensated for their costs in terms of CPU, network traffic, disk space and memory through fees that are proportional to the size (in bytes) of each transaction. However, each part of a transaction does not have an equal impact on the cost or the ability of Bitcoin to scale to support more transactions. The most expensive parts of a transaction are the newly created outputs, as they are added to the in-memory UTXO set, while the signatures (witness data) add the least load to the network and the cost of running a node, since witness data are only validated once and then never used again. Therefore, one idea is to discriminate between these two types of data relative to the burden a transaction imposes on the in-memory UTXO set. A proposal called **segregated witness (SegWit)**[26] has been put forward which separates transaction signatures from the transactions themselves. SegWit therefore has two main effects: it is able to both increase the number of transactions and to decrease the cost of transactions (fees).

4 The Bitcoin Application Point of View

Although the Bitcoin architecture was designed primarily for monetary transactions, it is also possible to extend it to smart contracts with some restrictions. The Bitcoin scripting language is also not Turing complete, and does not allow any algorithm running it to be extended beyond a payment infrastructure.

Since the Bitcoin architecture serves to keep internal transactions valid, it can also be used to confirm and validate specific, external, non-Bitcoin transactions. In other words, it enables the application of decentralized public ledgers for purposes other than digital currencies. Many companies such as IBM, SAP, Amazon and others are therefore

[24] http://lightning.network/lightning-network-paper-DRAFT-0.5.pdf
[25] https://www.blockstream.com/
[26] https://segwit.org/

experimenting with blockchain applications. These developments may disrupt the way people do business in the future.

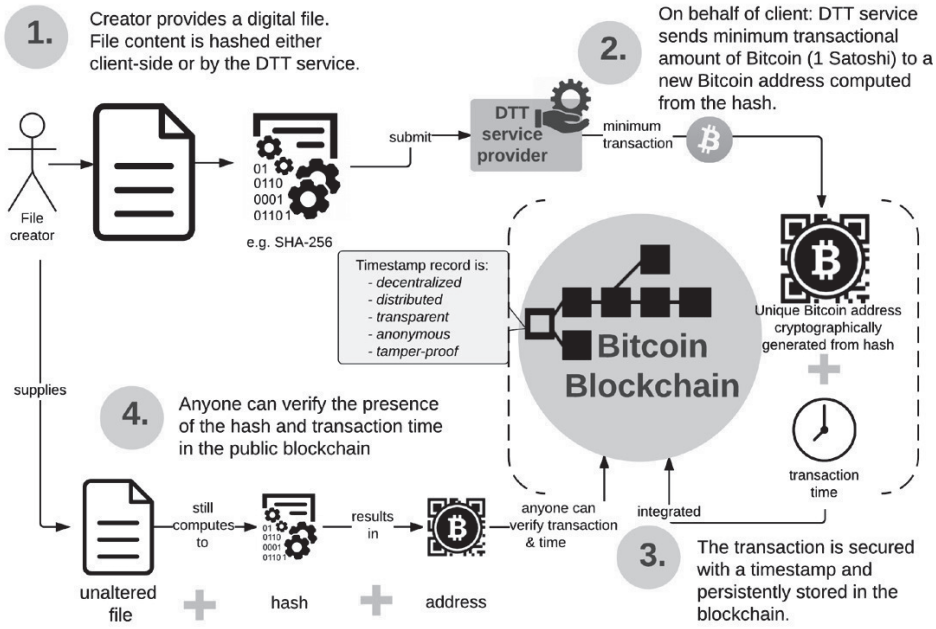


Fig. 11: The timestamping proof (www.originstamp.org)

One remarkable property of the blockchain is its **time stamp feature**. The whole network essentially validates the state of a wrapped piece of data (called a hash) at a certain specific time. The blockchain, hence, essentially confirms the existence of a record at a stated time, which is provable in a court of law. Until now, only centralized notary services could serve this purpose.

This proof of existence allows users to upload a file and pay a transaction fee to have a cryptographic proof (the hash of a document) included on the Bitcoin blockchain, although the file itself is not stored online and therefore does not risk unwanted publication (see Fig. 12). The document is hence timestamped using the block timestamp. Another application concerns **decentralized file storage** on the blockchain, which is likely to disrupt data storage and cloud computing within the next few years. Platforms like Sia[27] or Storj[28] use Blockchain technology to decentralize data storage by dividing up files, encrypting and transferring them all around the world. The accompanying

[27] https://sia.tech/
[28] https://storj.io/

cryptocurrencies (Siacoin, Storjcoin) incentivize usage and to create a market for decentralized storage.

Beside the Bitcoin blockchain there are several other platforms available for building distributed applications based on the blockchain technology. BlockApps, for example, is a good way to build decentralized blockchain-based applications. STRATO, BlockApps's[29] most prominent product, is a full-stack technology solution that allows users to build applications on top of their own customized permissioned private blockchain. This is built on the Ethereum blockchain which can be seen as a second generation blockchain with a built in programming language based on virtual machine concept.

Many of these applications are being currently developed, but the future potential of blockchain applications is still unfolding. The next few years will involve experimenting and applying this technology to all aspects of the economy. Regardless of which application comes first on a global scale. The blockchain is here to stay, and is transforming how business functions [DD17].

5 Conclusion

The blockchain technology first described in 1991 by Haber and Stornetta [HS91], in 1996 by Anderson [An96] and in 1998 by Schneier and Kelsey [SK98] got its sheer popularity through the first implementation of this idea as a distributed ledger called Bitcoin by Satoshi Nakamoto in 2008/09. Based on a simple scripting algorithm, the Bitcoin architecture promises secure and anonymous transactions for transferring money. However, the underlying blockchain technology has a much wider potential than simply transferring money.

As shown in this paper, transactions are first analyzed based on Bitcoin's protocol rules, and then validated using a scripting language based on locking and unlocking scripts. If a transaction is successfully validated, full Bitcoin nodes (miners) will select transactions from the memory-pool, hashing pairs of transaction values until only one hash remains (the Merkle root), and then anchoring the transaction tree into the block header. After choosing a nonce, the complete block header will be hashed and the result compared to a defined target with the PoW mechanism, as described above. If the hash value of the block header is less than the target, the block is mined, the rewards are paid out to the successful miner, and the transaction is included in the blockchain. After at least six blocks, a transaction can be considered irreversible.

Although the security model of Bitcoin promises complete privacy, it is not in general anonymous. As described above, the blockchain is public, meaning that anyone can see every Bitcoin transaction from the genesis block onwards. Bitcoin addresses cannot be

[29] https://blockapps.net/

directly associated to real-world identities though. Nevertheless, a great deal of work has been carried out on how to de-anonymize the Bitcoin network.[30] Furthermore, identification will be retrospective, meaning that someone who carried out a transaction in 2018 will still be identifiable on the basis of the block chain in, say, 2040.

Even if one does not believe in the specific cryptocurrency Bitcoin itself, the underlying blockchain technology will certainly survive and gain in relevance. In summary, Bitcoin, and especially the underlying blockchain system, is a very valuable technology from a scientific computer point of view (be it in computer science or other related fields such as information systems or organization studies), a thread for some businesses, and a gold mine for speculators [FH17].

6 References

[AA17] Antonopoulos, A.M.: Mastering Bitcoin, O'Reilly Media, Inc., 2017.

[AB02] Back, A.: Hashcash - A denial of service counter-measure, 2002.

[An96] Anderson, R.J.: The Eternity Service, Pragocrypt, 1996.

[DD17] Drescher, D.: Blockchain Grundlagen, mitp Verlag, 2017.

[DW13] Decker, C. and Wattenhofer, R.: Information propagation in the Bitcoin network, *IEEE P2P 2013 Proceedings*, 2013.

[FH17] Fraunhofer-Gesellschaft: BLOCKCHAIN UND SMART CONTRACTS Technologien, Forschungsfragen und Anwendungen, 2017.

[HS91] Haber, S. and Stornetta, W.S.: How to Time-Stamp A Digital Document, Journal of Cryptology, Vol.3, No.2, pp.99-111, 1991.

[JB04] Buchmann, J.: Introduction to Cryptography, 2004.

[RM80] Merkle, R.C.: Protocols for public key cryptosystems, In Proc. Symposium on Security and Privacy, IEEE Computer Society, 1980.

[Ro12] Rosenfeld, M.: Analysis of hashrate-based double-spending, arXiv:1402.2009, 2009.

[RW16] Wattenhofer, R.: The Science of Blockchain, Inverted Forest Publishing, 2016.

[SK98] Schneier, B. and Kelsey, J.: Cryptographic Support for Secure Logs on Untrusted Machines, in The Seventh USENIX Security Symposium Proceedings, pp. 53–62. USENIX Press, Januar 1998

[SN08] Nakamoto, S.: Bitcoin: A Peer-to-Peer Electronic Cash System, 2008.

[SW05] Steinmetz, R. and Wehrle, K.: Peer-to-Peer Systems and Applications, 2005.

[30] https://scholar.google.com/scholar?q=de-anonymization

Sicherheit

Müller, Neumair, Reiser, Dreo Rodosek (Hrsg.): 11. DFN-Forum Kommunikationstechnologien,
Lecture Notes in Informatics (LNI), Gesellschaft für Informatik, Bonn 2017 23

Stakeholder Specific Visualization and Automated Reporting of Network Scanning Results applying Vis4Sec

Tanja Hanauer[1], Stefan Metzger [2]

Abstract: This article introduces a process framework – Visualization for Security (Vis4Sec) – that supports the generation of organizational security knowledge and awareness. Vis4Sec is used to generate stakeholder specific visualizations based on the results of regular performed network scans in a complex IT infrastructure. The process steps Ask, Prepare Data, Visualize and Interact assist to define security relevant questions, prepare a data-driven visualization, embed it into an organizational context and distribute it. A proof of concept implementation was successfully done in a network environment operated by a Higher Educational Institution data center. Scan data resulting from several e. g. Network Mapper (nmap) based scanner machines has been aggregated and analyzed automatically, was then highly-enriched with organizational, security relevant information, visualized in dashboards, adapted to stakeholder specific requirements and distributed as reports.

Keywords: Information Security; Visualization of security-related data

1 Introduction

In many higher educational institutions' (HEIs) data centers IT systems are administered by different operating teams. In addition some services, e.g. hosting virtual machines or Infrastructure as a Service (IaaS) cloud services for institutions or individual users strengthen this and require the incorporation of customers' staff. This differs significantly from the centralized service operating models common in the industrial sector. HEIs also provide open and almost unrestricted infrastructures, so users are allowed to bring and connect their own devices or install any software found somewhere on the internet. There are no data center or network infrastructure wide asset management systems or configuration management databases, that provide a complete overview of the connected devices and the installed software. To cope with the security of such an ever changing environment the authors suggest the usage of integrated network scanning techniques, that provide an overview of IT systems, services, operating systems, and application software. The results of such regular scans and their deltas are a good starting point for security reporting. Unfortunately they are hard to grasp by the human eye as the results are in plaintext or in Extensible Markup Language (XML), which makes identifying security relevant changes difficult, especially when a huge number of systems are concerned. The visualization

[1] Leibniz Supercomputing Centre, Boltzmannstraße 1, Garching n. Munich, Germany hanauer@lrz.de
[2] Leibniz Supercomputing Centre, Boltzmannstraße 1, Garching n. Munich, Germany metzger@lrz.de

process Vis4Sec collects and aggregates these scan results at one central point and uses adequate data visualization methods as an approach to meet this challenge following the proverb *A picture is worth a thousand words*. Furthermore, existing organizational and security relevant information is correlated to handle and distribute those results best. The visualization process provides a framework for data acquisition, aggregation, visualization and organizational distribution of information. It supports the tracking of changes in the environment and makes the determination of the current attack surface possible.

The rest of this article is structured as follows: Section 2 provides a short overview of relevant security controls based on ISO/IEC 27001 and the Center for Internet Security Critical Security Controls (CSC), broaches data visualization techniques and introduces requirements of relevant stakeholders, before section 3 describes the Vis4Sec process itself. Section 4 shows a proof of concept process operation with the goal to limit and control open network ports. Section 5 summarizes the benefits of our approach and gives an outlook on future work.

2 State of the Art

This approach is based on actual security controls according to ISO/IEC 27001, good practices like the Critical Security Controls, well-known guidelines for data visualization, study-based stakeholder requirements analysis and and a brief selection of existing approaches.
The international standard ISO/IEC 27001 defines minimum requirements which an information security management system (ISMS) has to fulfill. Besides the general clauses provided in sections 4 to 10 of the standard, Annex A defines in total 114 reference security controls of which two are relevant for network scans. Control A.13.1.2 requires among other things inclusion and a regular review of technical and organizational security aspects related to network services. Control A.18.2.3 requires a technical review to ensure compliance with the organization's information security policies and standards. This encompasses the usage of tools, e. g. port scanners, or conducting penetration tests. While ISO/IEC 27001 does not require a specific implementation of security measures, the Critical Security Controls (CSC) [Ce16] expand on such details. The CSC are a widely used set of actions for cyber defense recommended by the Center for Internet Security. Controls with relation to automated network scanning are CSC 9 – Limitation and Control of Network Ports (9.1, 9.3), CSC 3 – Secure Configurations for Hardware and Software (3.6), and CSC 18 – Application Software Security (18.1, 18.4). CSC 9.1 and 9.3 are introduced in detail during the exemplary process run, the remaining controls are described briefly as subsequent iterations of the process. Subclause 9.1 ensures that only ports, protocols, and services with validated business needs are running on each system, which makes it directly mappable to controls in ISO/IEC 27001. Further relevant details are found in subclause 9.3 which requires automated regular port scans against all key servers and comparison of the results to a known baseline. This allows the discovery of unlisted changes to the organization's

approved baseline.
The generated visualization takes into account general knowledge about processing of visual information like the principles of Gestalt Theory, and design basics like Tufte's Design Criteria or Shneiderman's Information Seeking Mantra. Furthermore are challenges of information visualization in large companies addressed concerning the integration of tools in daily work processes, getting the data and being in constant close cooperation next to others as described by Sedlmair et al. [Se11]. The working conditions of system administrators, security personnel and members of the higher-level management described in qualitative ethnographic field studies have been analyzed and complemented with our own observations over a timespan of five years working in a HEIs' data center. Key findings according to Anderson [An02] are that transparency, notification, automation, schedulability, simplicity and scalability are very important criteria for admin tools. Besides that, data visualization can, as highlighted by Haber et al. [HK07] and Mahendiran et al. [MHZH12], provide improved system and security monitoring, a better overview of the current status of the infrastructure and simplicity of use for admins and security personnel. Human factors like the communication of security issues, organizational culture or an open environment are just like technical factors like the complexity of systems and applications and their vulnerabilities relevant for IT security management, as stated by Werlinger et al. [WHB08]. Current approaches in visualization research exist for security tasks like (network) security alert management [CvW16, FPLB17], network security and management [LS10], or even more specific topics, like visualization of ports or firewall configurations. These approaches are very task-specific and only focused on one use case for the decision support challenge at hand. Furthermore, a lot of visualization-method-specific research for security tasks exists with a prototypical implementation for one kind of visualization like assessing cyber incidents and network security with graphs [APS15], or ensembles [HHH15], or the visualization of specific data sources like data streams, web server and other log files. These approaches are helpful for the design of single visualizations, but they are also no solution for the organizational challenge at hand. Hence, the overall generation of visualization is taken into consideration described by visualization processes and frameworks like [Fr04, Ma08] and further developed into an integrated security specific management solution.

3 Process Vis4Sec

The analysis of stakeholder requirements, the collection of relevant data as a basis for the visualization, its organizational integration and the security requirements are implemented within a process framework – Visualization for Security (Vis4Sec). This framework offers a systematic approach to improve the information security of an organization. Figure 3 shows the iterative process Vis4Sec with its Initiation and four process phases Ask, Prepare Data, Visualize and Interact.

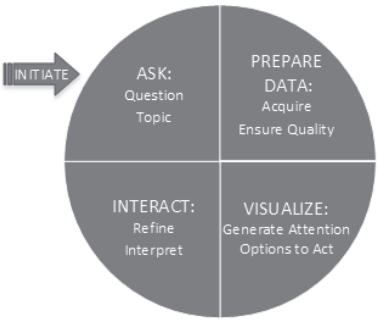


Fig. 1: Process Vis4Sec

Initiate: Vis4Sec starts by collecting and briefly describing necessary parameters for the visualization like the environment, the stakeholders, requirements and planned actions.

Ask: The initial question to be answered by the visualization has to be stated. It defines the context and it is the basis for the collection of relevant data. It is important to keep the question as simple as possible to allow for a successful process operation. In repeated iterations of the process the starting question is enhanced, refined or even completely redefined.

Prepare Data: This is by far the most extensive step as it consists of technical aspects like data collection, its pre-processing and analysis, and additional organizational aspects like the preservation, access control and disposal of data according to compliance and data protection regulations. It also includes the technical embedding into the organization like the definition of a data model, the insurance of data quality and the automation of the data collection and its quality insurance.

Visualize: The representation and presentation form for the quality ensured data is chosen with the goal to draw the attention of the recipient to the topic and offer options to act. The visualization should also generate awareness, provide a point of communication between the recipients and improve the status of the reported issue through the generated visibility.

Interact: The recipient provides feedback and adds expert knowledge that further improves the data visualization. Its utility is then evaluated according to the question asked during the question phase. New knowledge is interactively generated and the process starts over again with new or adapted questions derived from the previous result. New recipients obtain the improved or additional visualization.

Iterate: The process is iterated with modified questions stemming from the experience of the previous process runs. For example the specificity of the inquiry is enhanced, new data sources are added or existing ones are improved, and the feedback from the recipients of the visualization is built into the next process run.

4 Proof of Concept: Limitation and Control of Network Ports

In this section an exemplary process operation is shown as we deduce the question *"What are the reachable ports – externally or internally – on each system?"*. It is answered primarily by obtaining portscan data in the first run of the process. Afterwards data of the organization

is collected and its data quality is ensured using visualizations that track the data quality and initiate and accompany an organizational handling of the results. The aim here is to improve the organization's security especially towards the fulfillment of the requirements specified in CSC 9.1. In the next iteration of the process the requirements of CSC 18.1 are addressed and the false-positive rate decreased.

4.1 Initiation

The environment The Leibniz-Supercomputing Centre (LRZ) – a HEI's data center – providing more than 70 ICT and network services for institutions connected to the Munich Scientific Network (MWN), operates internally more than 130 server subnets to which more than 700 heterogeneous IT systems are connected. This environment can be specified as an overall complex and continuously changing setup. Usually, little knowledge about the concerned services and their probable vulnerabilities exists, which becomes obvious when security relevant questions are asked.

Requirements concern running services to be known, new services to be timely detected and that potentially vulnerable services to be recognized and patched as fast as possible. To fulfill those the distribution of stakeholder specific reports that provide an overview, impart knowledge and offer options to act, seems to be useful.

Stakeholder specific reports are designed for system administrators, security practitioners and IT management staff in this example scenario:
`Internal IT System Operations Teams (aka system administrators):` They configure, maintain and provide the IT systems and services. They are expert users with specific needs in terms of complexity, collaboration and risk. This group requires frequent reports with technical details about the configuration of the systems.
`IT Security Personnel (aka security practitioners):` Their focus lies on the security of the IT systems and the infrastructure. They have to deal with more complexity than system administrators, due to the high environmental rate of change, and the trade-off between usability, security and costs is also highly relevant. They occupy mostly a position in between since IT security often depends on the commitment of the management and the cooperativeness of the system administrators. This group requires frequent reports providing an overview and specific details to security related topics.
`IT Management Staff:` They are usually business-driven, and mainly responsible for making high level decisions. So they need abstraction from the technical aspects in form of information that allows them to make decisions based on correct data. This group requires infrequent reports in form of a high-level management view.

Planned Actions are the automation of network scans on a fine-granular level, followed by the collection and analysis of the scan results, their annotation with organizational and security-related information, and their stakeholder specific visualization and distribution, which lead to an enhancement of the organization's security level.

4.2 First Iteration: Open Network Ports on each System

Ask-1 According to the clauses in CSC 9 the question is: "What are the open ports, protocols and services with validated business needs running on each system?" To get a more practicable starting point the business need is left out, because it is not vital to answer in the academic environment of a HEIs' data center. Also the protocols and services are set aside for the next iterations and the question is thus simplified to: "What are the open ports on each system?" We will specify our question even more to further minimize the amount of results we get:[3]
"What are the reachable ports – externally or internally – on each system?"

Prepare Data-1 Resulting data from port scans is created, and organizational data from an already existing server configuration management database with detailed information about each system is collected and prepared. Furthermore, the quality of the data ensured. In the next iteration additional data, like the SSL configuration, is added to enrich the results.

`Data Source: Port Scanners` are tools easy to install and a simple network scan is also an easy task. But to do this in a more structured manner requires a comprehensive concept for deployment of several scanning machines, their operations and proper result processing. Answering questions about deployed scanning tools, usage of more than one scanning machine and their placement inside or outside the scanned network infrastructure, the scope of each scan performed and the repetition intervals is needed as Hommel et al. described [HSM15]. The complexity of this setup (several scanners, different locations) and various responsible contacts for the scanned subnets and machines bears the challenge of how to deal with the results, how to filter and distribute them among the stakeholders. This is done with the organizational data introduced in the following:

`Data Source: Organizational Data` is data that is almost static, e. g. basic information about the machines like the version of the operating system installed, its IP address, the system administrator's name and the department operating it. This information is extracted from a tool that functions as part of an organization wide Configuration Management Database (CMDB). It quickly became obvious that its data quality is unreliable, but this is crucial to provide useful results.

`The Data Quality` is ensured according to the six dimensions of data quality according to DAMA UK [ACea13]: completeness, uniqueness, timeliness, validity, accuracy and consistency. An overview of the data input's continuity is generated, quality checks of the organizational data are initiated, and an additional comparison with other data sources as data collected directly from the servers, in form of their installation base is also done in an iteration of the process.

Visualize-1 Visualization gives an overview or provides details if necessary. Dashboards in a WebGUI allow interactive usage of the information and PDF reports sent out by email

[3] The examples stem only from externally reachable systems.

present the information in a static form. Data is explored with use of an interactive dashboard as basis for the stakeholder specific dashboards and security enhancing visualizations. The first dashboard built functions as viewing tool. It is used to explore the data and extract areas of interest for stakeholder specific reports. The security practitioners are in most cases the user group that chooses the content of the visualizations because of their professional interest and their expert knowledge. The overview of the results from the port scans about the most exposed subnets or machines can support them to proactively enhance the overall security of the organization and in case of a newly disclosed vulnerability it provides them with the means to find the affected machines.
The interactive WebGUI makes it possible to filter the data and explore relevant subnets, groups or ports. The search capability allows to filter findings for one specific port, which is helpful to find machines providing a probably vulnerable service. So all systems providing services on that port and probably using the vulnerable product are quickly found and can be further investigated. For example in case of the Heartbleed OpenSSL flaw a search for servers providing OpenSSL services to the outside on port 443 was done on the Security Practitioners Dashboard shown in figure 2. It displays systems with an open web service

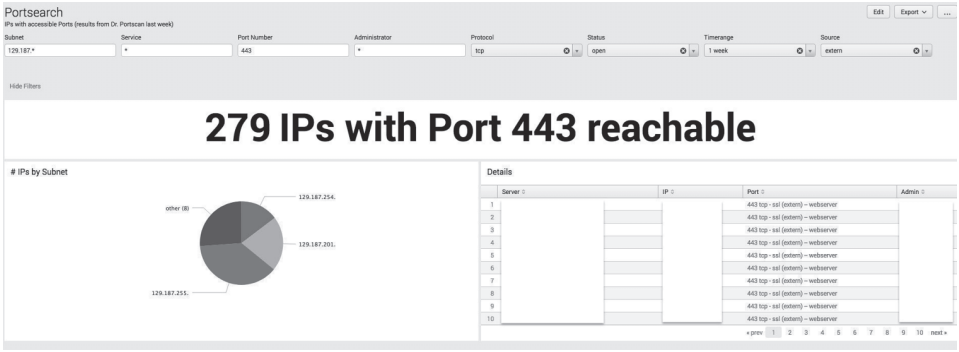


Fig. 2: Security Practitioners Dashboard: Search for web server systems on port 443 (HTTPS). The filters (top). The overall result (middle). The most exposed subnets (left). The details about each system (right).

on port 443 and the subnets where most of them are operated in. It can next to the port be filtered for subnet, service, system administrator, status (open/closed/filtered), time range or source (intern/extern). The pie chart on the left is used to highlight the most exposed subnets. It provides an easy orientation on which subnet to start with by representing the subnets with the highest number of IPs reachable on port 443.[4]

Interact-1: Before the security practitioner alerts the system administrators with a list of servers that *could be* vulnerable, further checks for the ciphers currently used and the actual exploitability of the service are initiated. The goal is to minimize the number of false positives and alert only, if there is a need for action. So the next iteration of the process operation with a redefined question starts.

[4] Additional ports like 22, 4443, 8443, . . . are handled the same way.

4.3 Second Iteration: Exploitable OpenSSL Library

Ask-2 The redefined question to answer is *"What are the externally reachable services that use an exploitable OpenSSL library?"*

Prepare Data-2 The detailed result from the search for systems with a reachable port 443 taken in the previous step is the data source at this point. But further data to identify a web server with a vulnerable OpenSSL version is necessary. This data is added once more as additional scan data – results of a SSL cipher-suite scan. The data quality of the results is ensured by comparison with the also newly added data source configuration data – package information from the servers directly, and information about the vulnerable and patched versions of OpenSSL – Common Vulnerability and Exposures (CVE) data.

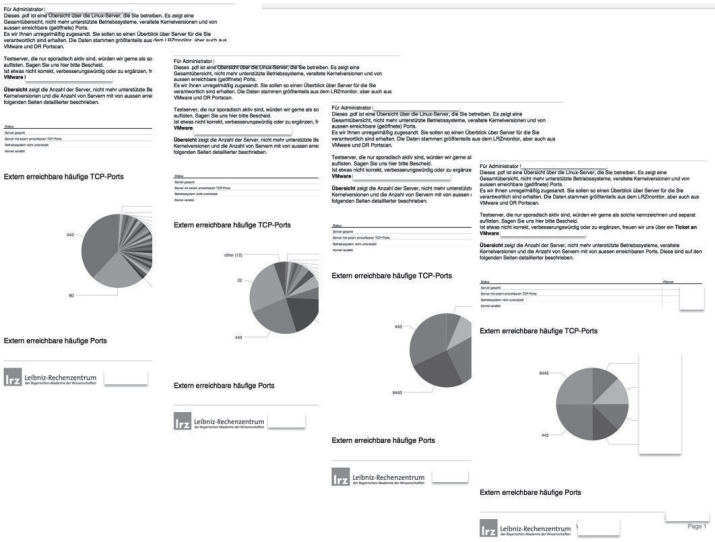


Fig. 3: Reports showing an overview of the most common services externally provided

Visualize-2 The second generated dashboard is a filtered version of the correlated data sources that functions as a request to act. It uses the data from the network scanners enhanced with the results from the cipher-suite scan, the organizational information and the configuration data and then adds a description of the vulnerability in question and how to fix it. For the system administrators it only displays the servers of a single system administrator that are externally reachable and also vulnerable to this exemplary attack. For the third stakeholder group – IT management –, that does neither need interaction with the data directly nor a lot of details, a dashboard providing an overall overview is generated and sent as a regular report. The results from the first iteration are processed to show the services in the organization. The derived report informs about the most common services and the groups providing those services. The second process iteration with OpenSSL results is used

as one of many data points to generate a quarterly report about the patch status and the reaction to actual vulnerabilities.

Interaction-2 Finally all of these dashboards and the searches behind are adapted to fulfill stakeholder specific requirements, from which reports are automatically generated and sent with ReMailS (Report Mailer for Splunk). The reports are generated on configurable intervals and they are embedded into an organization wide feedback system. The first pages of exemplary reports sent regularly to single system administrators and the IT management in an aggregated form are shown in figure 3.

4.4 Further Iterations

In a further iteration the interactivity of the dashboard is enhanced and a search is provided that enables security staff to search ad hoc for vulnerabilities. This was described for a web service on port 443, but in the same iteration it was also done for SSH (22) providing externally reachable management access, which is often an unnecessary exposure of a service.

5 Conclusion and future research

The visualization of security-related data using the Vis4Sec framework is a trigger for security enhancement in an organization. It provides a framework to track and continuously improve the security level of different areas. Based on simply obtainable data like results of network scans correlated with other data sources, the security level of the application software or the compliance to data protection regulations can be ensured. The iterative process approach ensures stepwise refinement of the questions and results meeting stakeholders' needs and the focus on feedback improves the quality of the data, generates organizational knowledge and communication points. Further iterations with refined questions like "Are the software versions used still supported by the vendor?" (CSC 18.1) stemming from the ISO/IEC 27001 and CSC are planned. A further process iteration with data on software packages, CVE data including Common Vulnerability Scoring System scores is in preparation. Also an iteration asking 'What are new listening ports, new administrative users, changes to groups and local policy objects or new services running on a system?" (CSC 3.6) sounds promising.

References

[ACea13] Askham, Nicola; Cook, Denise; et al., Martin Doyle: The Six Primary Dimensions for Data Quality Assessment. White paper, Data Management Association UK, October 2013.

[An02] Anderson, Eric Arnold: Researching system administration. Phd, University of California at Berkeley, 2002.

[APS15] Angelini, M.; Prigent, N.; Santucci, G.: Percival: proactive and reactive attack and response assessment for cyber incidents using visual analytics. In: IEEE Symposium on Visualization for Cyber Security (VizSec). pp. 1–8, Oct 2015.

[Ce16] Center for Internet Security: , The CIS Critical Security Controls for Effective Cyber Defense Version 6.1, August 2016.

[CvW16] Cappers, Bram C. M.; van Wijk, Jarke J.: Understanding the context of network traffic alerts. In: 2016 IEEE Symposium on Visualization for Cyber Security, VizSec 2016, Baltimore, MD, USA, October 24, 2016. pp. 1–8, 2016.

[FPLB17] Franklin, L.; Pirrung, M.; L. Blaha, et al.: Toward a visualization-supported workflow for cyber alert management using threat models and human-centered design. In: 2017 IEEE Symposium on Visualization for Cyber Security (VizSec). pp. 1–8, Oct 2017.

[Fr04] Fry, Benjamin Jotham: Computational Information Design. PhD thesis, Massachusetts Institute of Technology, April 2004.

[HHH15] Hao, L.; Healey, C. G.; Hutchinson, S. E.: Ensemble visualization for cyber situation awareness of network security data. In: 2015 IEEE Symposium on Visualization for Cyber Security (VizSec). pp. 1–8, Oct 2015.

[HK07] Haber, Eben M; Kandogan, Eser: Security Administration in the Wild: Security Administration in the Wild: Ethnographic Studies of Security Administrators. In: ACM SIG CHI. 2007.

[HSM15] Hommel, Wolfgang; Stefan Metzger, et al.: Improving higher education network security by automating scan result evaluation with Dr. Portscan. In: EJHEIT. volume 2 of EUNIS 2014 – 20th EUNIS Congress, Umeå, Sweden, pp. 11–20, May 2015.

[LS10] Liao, Qi; Striegel, Aaron, et al.: Visualizing Graph Dynamics and Similarity for Enterprise Network Security and Management. In: Proceedings of the Seventh International Symposium on Visualization for Cyber Security. ACM, NY, USA, pp. 34–45, 2010.

[Ma08] Marty, Raffael: Applied security visualization. Addison-Wesley, cop, http://www.conkel.net/download/Applied2008.

[MHZH12] Mahendiran, Jeevitha; Hawkey, Kirstie; Zincir Heywood, Nur: Understanding the Use of Models and Visualization Tools in System Administration Work. Dalhousie University DCSI Proceedings, 2012.

[Se11] Sedlmair, Michael; Isenberg, Petra; Baur, Dominikus; Butz, Andreas: Information visualization evaluation in large companies: Challenges, experiences and recommendations. Information Visualization, 10(3):248–266, 2011.

[WHB08] Werlinger, Rodrigo; Hawkey, Kirstie; Beznosov, Konstantin: Human, Organizational and Technological Challenges of Implementing IT Security in Organizations. In (Clarke, Nathan L.; Furnell, Steven, eds): HAISA. University of Plymouth, pp. 35–47, 2008.

Performance Evaluation in High-Speed Networks by the Example of Intrusion Detection Systems

Thomas Lukaseder[1] Jessika Fiedler[1] Frank Kargl[1]

Abstract: Purchase decisions for devices in high-throughput networks as well as scientific evaluations of algorithms and technologies need to be based in measurements and clear procedures. Therefore, evaluation of network devices and their performance in high-throughput networks is an important part of research. In this paper, we document our approach and show its applicability for our purpose in an evaluation of two of the most well-known and common open source intrusion detection systems, Snort and Suricata. We used a hardware network testing setup to ensure a realistic environment and documented our testing approach. In our work, we focus on accuracy of the detection especially dependent on bandwidth. We would like to pass on our experiences and considerations.

Keywords: IDS; performance tests; high-speed networks

1 Introduction

Computer networks provide a wide range of targets for malicious purpose. An attacker can be able to steal sensitive data or even bring the system to a standstill. For this reason, there must be mechanisms to protect networks. Intrusion detection systems (IDS) are one of the most important components of a modern security infrastructure. Especially communities and companies that deal with a large and growing amount of data have increasing demands on current computer networks. This is the reason why 10 Gbps Ethernet has become a standard within company and university networks. In terms of security, IDS are commonly used to inspect networks and as an essential component to find different kinds of attacks. As they observe network traffic, network-based systems are now facing a great challenge and it must be tested, whether they can handle it. There is a clear need for an overview of the different systems and their performance in high-speed network traffic, so provider get an idea of what solution to choose. Such an overview is challenging as the term Intrusion Detection System is used for a wide area of technologies. Therefore it must be discussed how those different solutions can be evaluated in a comparative way. As the results are supposed to be comparable, it must be ensured that all of the solutions run under exactly the same conditions. This means, they should get the same resources in terms of hardware and input data. Based on our previous work in building network testing frameworks [BLK17][Lu16b] and testing

[1] Universität Ulm, Institut für verteilte Systeme, Albert-Einstein-Allee 11, 89081 Ulm, Germany, firstname.lastname@uni-ulm.de

networks [Lu16a], we evaluate IDS in high-throughput networks. The paper is organised as follows: Section 2 lists comparable works our work is based on, Section 3 elaborates on the IDS we chose to evaluate and why while Section 4 documents our approach. In Section 5 the most important evaluation results are summarized. Section 6 concludes the paper.

2 Related Work

There is a lot of work discussing IDS and their ability to handle high-speed traffic. Most of them focus on improving the settings of software and systems to the new requirements. When it comes to performance evaluations in high-speed networks, there are only a couple of detailed reports available. Concerning the general evaluation of IDS, Milenkoski et al. [Mi15] should be mentioned. The authors give a detailed overview of practices, that are used to evaluate IDS. They structure the evaluation into three parts: workloads, metrics and methodology. We use this as a guideline for this project and help to generate benchmark data. In the last part they present measurement methodologies, which define the systems properties of interest as well as the employed workloads and metrics to evaluate this properties. However, they do not conduct any tests. The three open-source solutions of Intrusion Detection Systems — namely Suricata, Bro and Snort — have been tested extensively. Those tests mostly focus on their accuracy. Only a few of them evaluate their performance in high-speed networks. Khalil [Kh15] gives a good overview of the three open-source solutions. He outlines their problems with handling high-speed traffic and the solutions those systems implement. He also outlines a couple of performance tests that have been run on those systems. They analyzed overall performance concerning traffic throughput but did not analyze the precision of the IDS under test. Bulájoul et al. [BJP13] ran a couple of tests to evaluate Snorts performance. For their analysis they altered three different values: the number of packets per second, number of packets sent over all and the packet length. As performance indicators they use the number of packets received, analyzed, dropped, and filtered. The number of packets filtered does not occur in the figures, only the number of packets sent. According to the authors this value is not altered during the experiment, only the speed at which those packets have been sent. There is a figure included where the number of packets sent decreases when the speed increases. This is contrary to the point that the number of packets sent is not altered. Furthermore, this value should be redundant to the number of packets received as Snort should receive the packets sent, however, the data indicates otherwise. Nevertheless, their experiments show that the number of packets dropped increases, when they increase the speed. This could give a basic indicator of Snorts performance in high-speed traffic. Most IDS answer the challenge to handle high-speed networks with some kind of load-balancing. Vallentin et al. [Va07] present a cluster solution for network-based intrusion detection. They introduce Cluster Bro, which is a solution to handle high-speed traffic with Bro. The input is split across several 'worker' nodes, that analyze their part of the traffic and a management system is used to perform the overall analysis. Their work also includes a performance evaluation to show that the cluster produces sound results even at high throughput. They also mention that their

solution ran at IEEE Supercomputing 2006, where it 'monitored the conference's primary 1 Gbps backbone network as well as portions of the 100 Gbps High Speed Bandwidth Challenge network which is a very broad term and could mean anything between 0 and 100 Gbps. The basic points are that the cluster gives the same results as a single Intrusion Detection System. In their performance evaluation they focus on the ability to balance the incoming traffic to the different nodes — namely the scalability — and the overhead of communication this cluster solution introduces in comparison to a single IDS. There is no evaluation of the Cluster's accuracy as part of this paper.

3 Systems under Review

There are three relevant open-source IDS that can be found in related work — Bro, Suricata, and Snort. We limit our analysis to Snort and Suricata as Bro is an anomaly-based IDS that functions differently to the other two in many ways. Snort[2] is an open-source network-based Intrusion Detection System. Its functionalities include packet capturing, analysis of captured packets and — important for this work — live analysis of network traffic. The version tested in this paper is 2.9.9. A lot of dependencies have to be installed before Snort is able to run properly. Those include libnet — a network API to gain access to different protocols, libpcap — a library to capture network traffic and pcre — a collection of functions with Perl semantic and syntax. New to version 2.9 is the usage of the DAQ library, which replaces libpcap calls to simplify packet-I/O-options.
Snort provides the possibility for developers and users to add own modules called preprocessors. One of them is Perfmon [St09]. Using this preprocessor the user gets informed about statistics such as received and dropped packets during runtime. To use the preprocessor a corresponding option has to be set during the build process.
After Snort has been installed successfully, it needs to be configured. This is done using the configuration file written in a specific Snort format. For this work, configuration meant to set network variables, choose active rules, activate the Perfmon preprocessor, and set output formats. There are three different sets of rules: subscriber, registered, and community rules. The basic set of rules is taken from the subscriber snapshot and is extended by rules to detect included attacks.
Another open-source network-based Intrusion Detection System is Suricata[3]. Its version 3.2.1 is the basis for testing. Besides the same dependencies as Snort, it needs libraries to process files in yaml format. This is used to read and write the configuration file. Different from Snort is the multi-threading option. This is included in all Suricata versions and needs to be configured correctly. Its performance is depending on the number of processors and some memory caps that have to be set properly with regards to the chosen number of processors. Fine tuning of Suricata is done according to [Re]. During a first experiment, it became obvious that the number of processors influences the performance, and with more processors memory consumption grows extremely.

[2] snort.org
[3] suricata-ids.org

Suricata is able to parse and use Snort rules. In addition, there is a set of rules called emerging rules [Su], which has been used as a basis. During live analysis, Suricata produces alert logs in the same format as Snort. Furthermore, Suricata outputs statistics on the current values of received and dropped packets.

4 Evaluation Setup

Two networks have to be simulated: an external network and a home network with the IDS in the middle. Two server are used to simulate the networks. One of them to send malicious and benign packets and the other to receive them. Both are connected via 10 GB interfaces to the IDS server. The server has to have sufficient resources to be capable of analyzing the traffic at line speed. We chose an off-the-shelf server with 4 CPU cores with 3.1 GHz[4] and 6 GB of memory; multi-threaded intrusion detection systems can make use of 4 threads per processing step.

4.1 Traffic Simulation

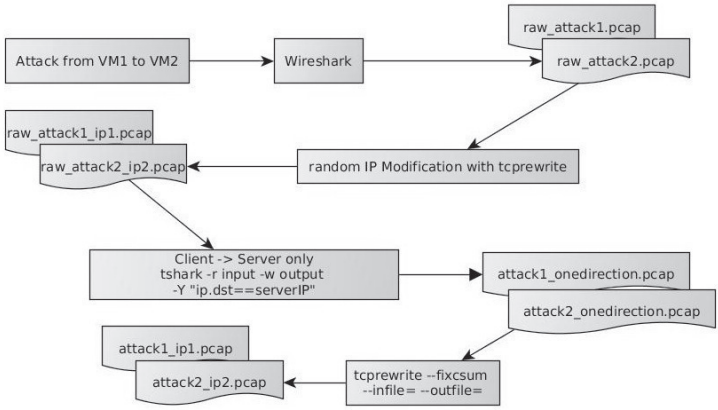


Fig. 1: Generating malicious traffic captures.

Malicious traffic is simulated and captured before the actual evaluation so repeatability of the tests is ensured. The attacking VM is running Kali Linux, whereas the target machines run Ubuntu Server, one with Metasploit. Wireshark is used to capture traffic. Target IP addresses are set to the receivers incoming interface. The attacking IP address is changed in a manner that each attack has its own sending IP address. This is done to distinguish logged alerts. As the response of the receiving server is not analyzed by the IDS under test, they

[4] IntelXeon Processor E3-1220 v3

are removed from the captured traces. Figure 1 shows the complete process of generating attack captures.

We used the threat report by McAfee Labs[Mc16] to determine a typical attack mix that realistically represents the day to day threats of networks connected to the Internet. According to the report, browser-based attacks are the most prevalent, followed by brute force login attempts (e.g. SSH brute force), followed by denial of service attacks, SSL attacks, scans (e.g. nmap scans), DNS spoofing, and backdoors. All in all, these attacks account for 91% of all network attacks. We generated these attacks with different tools. The whole list can be seen in Table 1. The benign traffic is simulated with iperf3[5], as it is capable of generating 10 Gbps traffic at runtime mixed with data peaks. To simulate a data center it is necessary to include data peaks. Large datasets are publicly available at different sources (KDnuggets [KD] as an example). Those captures can later be resent several times per evaluation run. A problem is to gain appropriate traffic speed with tcpreplay, this could only be achieved when storing the data set in a RAM-disk.

4.2 Rule Selection

Appropriate rules for the IDS must be chosen to ensure that they are able to detect all attacks included in the attack traffic — given that the IDS has the time to analyze the traffic. Furthermore, the traffic has to be labeled to later decide if a detection is a false or true positive. The rule sets for the IDS under test follow different syntax but have to be semantically identical to ensure that speed differences between the IDS only depend on the IDS implementation. For the performance evaluation a minimum set of rules has been chosen as the purpose is to measure the capability to handle high-speed traffic not the influence of the number of rules. Thresholds for detection and similar parameters are chosen identical for both systems (e.g. 150 as the threshold for flooding attack alerts). The number of active rules influences the overall performance. Therefore, this number is the same for both IDS.

[5] iperf.org

Attack type	Tool used
successful SSH brute force	Metasploit framework
unsuccessful SSH brute force	Metasploit framework
TCP connect flood	nping
TCP SYN flood	hping3
UDP flood	hping3
SYN scan	nmap -sS
SYN OS-scan	nmap -sS -O
UDP scan	nmap -sU
User enumeration	nmap

Tab. 1: Included attack types.

Program part	IDS	Sender	Receiver
Start	start IDS	read in attacks	
	wait for IDS		
Monitoring	CPU, memory, traffic	outgoing traffic	incoming traffic
Evaluation part	wait	send packets	wait
Output	modify output files		
End	kill IDS instance		
	Resting phase		

Tab. 2: Per server tasks during the evaluation program.

4.3 Evaluation Process

A single test needs four parameters: the time for the evaluation, the number of attacks per minute, the traffic speed, and the chosen IDS. During one test, three phases can be distinguished: initialization, evaluation, and output. During the first step the IDS is initialized and the attack plans for the setup are read from configuration files. Those files contain a list of attacks that should be send at minute m during an evaluation of M minutes. This ensures that each IDS is tested with the same attacks and that there are no variations within a test run. During the test run, traffic, CPU, and memory usage is monitored. Benign traffic is sent continuously at the given speed and the defined attacks per minute using tcpreplay. In this setup, more attacks also results in a slightly higher number of packets per minute. Table 2 shows the steps on each server during a single test. The output of a certain throughput and different amount of attacks is one test sample (e.g. 5,10,15,20 attacks at 2 Gbps). A test phase includes all test samples for a chosen IDS. This means tests are performed for throughputs and attack amounts in given ranges. A sample test and a complete test phase are both completely automated. The processes on all three servers must be started at the same time, synchronized clocks on all servers are required.

4.4 Result Processing

Each test has several outputs in different formats. The result processing for a single test — meaning exactly one value for speed and one for attacks per minute — are discussed first. Timestamps must be compared to ensure that all outputs are within the same time period. Most traffic speed log files differ in their units. They are recomputed to Gb and the different bandwidth files are combined to one. This can be used to check the resulting traffic throughput during evaluation and find bottlenecks if those exists. As mentioned earlier, all included IDS output information about their current received and dropped packets. Those values are extracted and combined with the measured CPU and memory usage. The CPU documentation gives the amount of CPU usage per core. Due to some rule settings, it is possible that a reference log file includes messages that are not necessary to identify an attack. This especially occurred when detecting SSH brute force attacks with Suricata.

Value	Meaning	Arithmetic
True Positive TP	Correct logged messages	sample sum
False Positive FP	Logged but not expected	sample sum
False Negative FN	Expected but not logged	sample sum
True Positive Rate	Attack detection rate (Sensitivity)	TP/(TP + FN)
False alarm rate	Rate of false alarms	FP/(TP + FP)
Precision	Rate of correct alerts among all alerts	TP/(TP + FP)
CPU	CPU usage of IDS	sample average
Memory	Memory usage of IDS	sample average
Received packets RP	Packets analyzed by IDS	average over time
Droprate DP	Packets dropped by the IDS	average over time
Send packets SP	Actual send packets	average over time

Tab. 3: Result values per test sample.

During the reference phase, logs contain messages such as 'ET INFO NetSSH SSH Version String Hardcoded in Metasploit'. Those are legitimate messages but not necessary to identify the actually attack — however, the message 'ET SCAN Potential SSH Scan' is. To keep track of messages that can appear but do not have to, priority files have been introduced. They include the attack message and a priority — 0 means the message must be there and 1 it is acceptable if it is missing. The value is only added to false positives if the message itself really was not expected in this minute. Some messages are redundant. As an example Snort either logs a 'TCPFilteredScan' or 'TCPScan' message when detecting a SYN scan. Those messages are mapped so they end up as correctly logged alerts. The result of the process is a file containing a mapping of minute, message to logged- and expected counters as well as a statistic counted per type of attack. The first file can be used to identify the (mis-)matches per minute during one test. Furthermore, those values are used to identify the detection performance. Once all tests of a test sample have been processed, sample wide values can be counted. This means per minute attack detection statistics are summed up and used to compute performance measurements. One of them is true positive rate or sensitivity which represents the detected attacks out of all attacks. Another value of interest is precision giving the percentage of correctly identified alerts among all logged alerts. CPU and memory consumption are averaged over all measured values. Packet information is taken from the IDS performance output and from system log files. Suricata outputs totals of the current values, Snort averages over runtime, therefore values are computed into averages over runtime for all packet information files. The same has to be done for alerts per second values and the drop rate. The number of packets that have not been considered can be computed from the number of packets that have been send and the number of packets that have been analyzed by the IDS. All output values for a test sample are summarized in Table 3.

5 Results

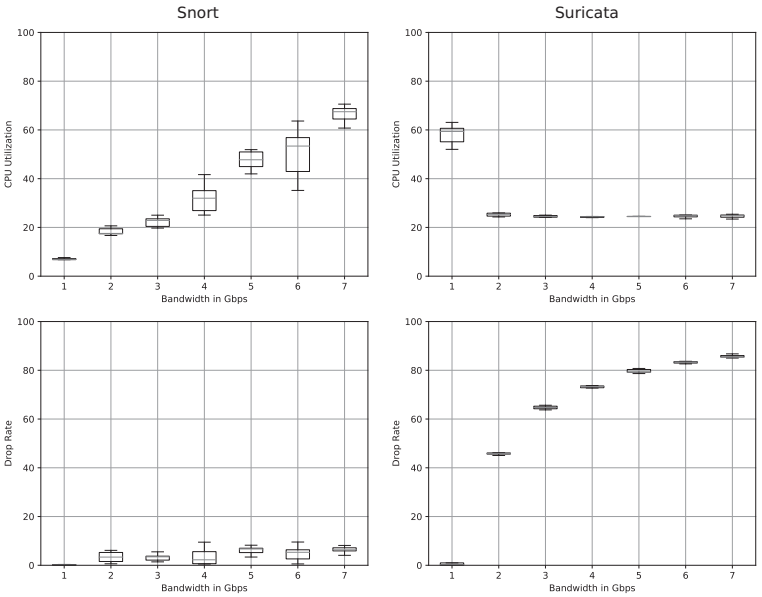


Fig. 2: CPU utilization per CPU core in use and packet drop rate of Snort and Suricata dependent on bandwidth.

The evaluation part included tests for 30 minutes each with equal attack distributions. Meaning all types of attacks mentioned previously are included with the same probability. Traffic speeds from 1 to 7 Gbps have been included as well as attacks from 10 to 35 per minute. The data implicates, that throughput has great influence on the CPU utilization and drop rate while the amount of attacks does not influence these factors. Precision and sensitivity on the other hand where not influenced by the throughput (despite the high drop rate differences) but only by the amount of attacks in the network. Neither bandwidth nor amount of attackers had influence on memory usage (Snort: 6MB, Suricata: 80MB). There were no false positives observed in any test scenario.

Figure 2 shows the CPU and packet drop measurements. Every data point contains measurements with different amount of attackers (10, 15, 20, 25, 30, and 35 attackers per minute). While Snort's drop rate is low throughout all measurements, the CPU utilization rises linearly and reaches up to 65% with 7 Gbps of traffic. Suricata's approach is different. The CPU utilization per CPU core stays quite constant other than the measurement with only 1 Gbps of traffic as less CPU cores are in use here. However, Suricata begins to drop packets as soon as the throughput is higher than 1 Gbps. Suricata consistently analyses 1 Gbps and drops any traffic exceeding that limit.

Figure 3 shows the precision and sensitivity measurements. Every data point contains

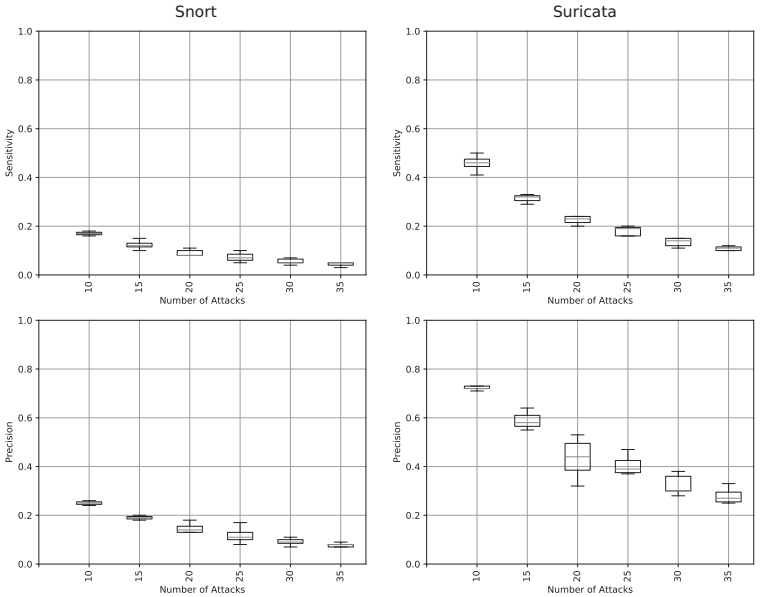


Fig. 3: Precision and sensitivity dependent on the amount of attacks per minute.

measurements with different throughputs of attack traffic (1, 2, 3, 4, 5, 6, and 7 Gbps). Although Suricata heavily drops packets depending on throughput, and Snort does not, higher throughput did not influence the precision or sensitivity. However, more attacks in the test traffic did lead to worse detection rates. All in all, Suricata showed better performance.

6 Conclusion and Future Work

In this work, we described a possible way to handle performance evaluation tasks and used this process to evaluate two software-based IDS and stress tested their performance. We used a hardware-based network setup with commodity hardware, tested with different throughput and different amounts of attacks per minute to evaluate which parameters influence performance the most and what administrators have to keep in mind when choosing and setting up an IDS in their network. We showed that bandwidth only plays a minor role for the precision of the systems under review, while the amount of expected attack traffic should be considered. Furthermore, the systems under review tend to heavily rely on good CPU performance while their memory requirements are comparably low.

For the future, we will extend our evaluation to other IDS (e.g. Bro) and to test beyond 10 Gbps on stronger hardware to evaluate whether software IDS can be applicable in backbone networks. We plan to update the data set regularly based on the threat reports

and — together with the configuration files for the IDS — publish this data. Furthermore, we plan to extend the evaluation work to include other security devices such as firewalls.

Acknowledgment

This work was supported in the bwNET100G+ project by the Ministry of Science, Research and the Arts Baden- Württemberg (MWK). The authors alone are responsible for the content of this paper.

References

[BJP13] Bulajoul, W.; James, A.; Pannu, M.: Network Intrusion Detection Systems in High-Speed Traffic in Computer Networks. In: 2013 IEEE 10th International Conference on e-Business Engineering. pp. 168–175, Sept 2013.

[BLK17] Bradatsch, L.; Lukaseder, T.; Kargl, F.: A Testing Framework for High-Speed Network and Security Devices. In: 2017 IEEE 42nd Conference on Local Computer Networks (LCN). pp. 506–509, Oct 2017.

[KD] KDnuggets: , KDnuggets: Datasets for Data Mining and Data Science [Homepage]. http://www.kdnuggets.com/datasets/index.html.

[Kh15] Khalil, George: Open Source IDS High Performance Shootout. SANS Institute, 2015.

[Lu16a] Lukaseder, T.; Bradatsch, L.; Erb, B.; Heijden, R. W. Van Der; Kargl, F.: A Comparison of TCP Congestion Control Algorithms in 10G Networks. In: 2016 IEEE 41st Conference on Local Computer Networks (LCN). pp. 706–714, Nov 2016.

[Lu16b] Lukaseder, T.; Bradatsch, L.; Erb, B.; Kargl, F.: Setting Up a High-Speed TCP Benchmarking Environment - Lessons Learned. In: 2016 IEEE 41st Conference on Local Computer Networks (LCN). pp. 160–163, Nov 2016.

[Mc16] McAfee Lab Quaterly Thread Report. http://www.calyptix.com/top-threats/top-7-network-attack-types-2016/, Accessed: 2018-01-24.

[Mi15] Milenkoski, Aleksandar; Vieira, Marco; Kounev, Samuel; Avritzer, Alberto; Payne, Bryan D: Evaluating Computer Intrusion Detection Systems: A Survey of Common Practices. ACM Computing Surveys, 48(1):1–41, 2015.

[Re] Regit - Suricata, to 10Gbps and beyond [Blog post]. https://home.regit.org/2012/07/suricata-to-10gbps-and-beyond/, Accessed: 2018-01-24.

[St09] Sturges, Steve: Using Perfmon and Performance Profiling to Tune Snort Preprocessors and Rules. 2009.

[Su] Suricata - Emerging ruleset. https://rules.emergingthreats.net/open/suricata/rules/, Accessed: 2018-01-24.

[Va07] Vallentin, Matthias; Sommer, Robin; Lee, Jason; Leres, Craig; Paxson, Vern; Tierney, Brian: The NIDS Cluster: Scalable, Stateful Network Intrusion Detection on Commodity Hardware. Proceedings of the 10th international conference on Recent advances in intrusion detection, pp. 107–126, 2007.

OFFWall: A Static OpenFlow-Based Firewall Bypass

Bastian Germann[1] Mark Schmidt[2] Andreas Stockmayer[3] Michael Menth[4]

Abstract: Stateful firewalls are becoming bottlenecks for high-speed communication networks. To counteract, trusted network flows may statically bypass the firewall. As access control lists (ACLs) of moderately priced switches do not allow port selection, they cannot be used for implementation of a static firewall bypass. In this work, we present a software-defined networking (SDN) based solution for a static firewall bypass based on moderately priced commodity hardware. We propose OFFWall, an OpenFlow (OF) controller that translates a whitelist of trusted flows into flow rules and installs them on an SDN switch to implement the firewall bypass.

OFFWall has been developed according to the demands of network administrators. Its goal is simplicity and stability so that it can run for long time without updates. Therefore, it is programmed in Rust for runtime stability and compiled to an executable file. Moreover, it offers only a minimal feature set required to install and remove flow rules on the switch.

After successful tests in a virtual and physical setup with different complexity, we deployed OFFWall on the network of the Department of Computer Science of the University of Tuebingen.

Keywords: SDN; OpenFlow; firewall; bypass; whitelist; traffic flow; Rust

1 Introduction

With increasing transmission rates in communication networks, firewall clusters at moderate cost become bottlenecks. Diverting some traffic around the firewall can alleviate the bottleneck effect. Network traffic that is considered secure, e.g., internal traffic that is forwarded to a different net within the same organisation, can be bypassed, which we refer to as "firewall bypass."

Fig. 1 depicts the general concept of a firewall bypass. The router and the switch are the edge devices between two networks. The switch is connected to a firewall. A packet from the switch's outside port (1) is forwarded to the firewall-out port (2) by default. After firewall

[1] University of Tuebingen, Chair of Communication Networks, Sand 13, 72076 Tuebingen, Germany
bastian.germann@student.uni-tuebingen.de
[2] University of Tuebingen, Chair of Communication Networks, Sand 13, 72076 Tuebingen, Germany
mark-thomas.schmidt@uni-tuebingen.de
[3] University of Tuebingen, Chair of Communication Networks, Sand 13, 72076 Tuebingen, Germany
andreas.stockmayer@uni-tuebingen.de
[4] University of Tuebingen, Chair of Communication Networks, Sand 13, 72076 Tuebingen, Germany
menth@uni-tuebingen.de

processing, it is received at the firewall-in port (3) and output to the switch's inside port (4). A packet from the inside network is forwarded in the opposite direction by default. The basic idea of a firewall bypass is a shortcut for trusted network flows from the outside port to the inside port of the switch and in the opposite direction.

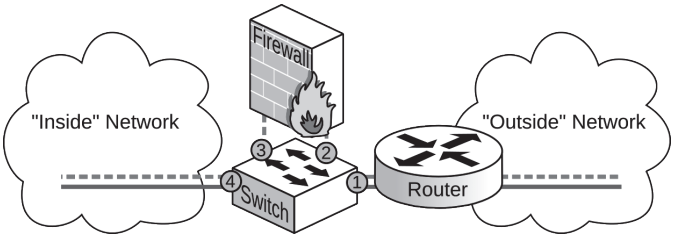


Fig. 1: A firewall bypass.

A *static* firewall bypass uses static whitelist rules in the switch to define network flows that are bypassed. All other flows take the default route through the firewall. A whitelist rule maps a 5-tuple (source IP address, destination IP address, protocol, source port, destination port) plus the incoming switch port to the intended output switch port.

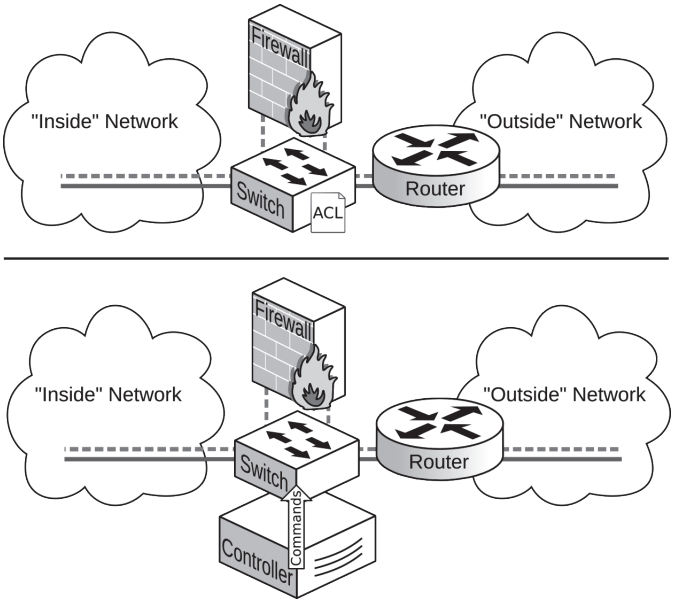


Fig. 2: A firewall bypass can be implemented via ACLs or SDN.

Fig. 2 depicts two possible ways to specify whitelist rules for network flows. One way involves a switch's built-in ACL. We found the ACLs of different Layer-3 switch devices to be insufficient for this task. E.g., the edge device used at our campus network does not

support port selection of outgoing packets via ACLs. Devices that have this feature are much more expensive.

In this paper, we present a firewall bypass based on SDN instead of ACLs. A switch can be utilised to implement a firewall bypass if it supports an SDN protocol to manipulate its network flows. In the SDN-based firewall bypass, the default traffic flow and the firewall bypass rules are installed on an SDN-capable switch by an SDN controller that is run in a separate management network. This approach is a cost-effective alternative to acquiring a device that provides sophisticated enough ACLs for this problem.

The most widely used SDN protocol that addresses manipulating network flows is OF. It is available in many Layer-3 switch devices. OF switches are programmed by an OF controller. The rules controlling the network flow are called flow entries. There are several OF controller frameworks which offer a rich feature set but lack runtime stability that is desirable for continuously operating an OF controller for an edge device. We observe that for a firewall bypass only a small subset of the OF specification is utilised on the controller side to install flow entries.

Therefore, we propose OFFWall [Ge] which is a functionally reduced OF controller. This controller implements just enough OF specifics to proactively install flow entries on an OF switch. All of these specifics belong to the required OF feature set. This approach allows for a small amount of source code in our implementation that runs solidly and is easy to reason about.

The rest of the paper is structured as follows. Sect. 2 reviews related work on firewall bypassing and OF. Sect. 3 elaborates on the overall architecture and implementation of OFFWall. In Sect. 4, we validate the achievement of OFFWall's design goals with tests and present the final deployment. Sect. 5 concludes this work.

2 Related Work

In this section, we first present an intuitive Layer-3 based concept for a firewall bypass that we discarded for practical reasons. Then we introduce background knowledge of SDN and OF. Finally, we report on dynamic firewall bypasses that have been considered in scientific work and projects.

2.1 A Router-Based Static Firewall Bypass

Given, the Cisco ASA firewalls in use at our campus network can be operated in transparent or routed mode (Chapter "Transparent or Routed Firewall Mode" in [Ci17]), there is another intuitive, valid approach to firewall bypassing.

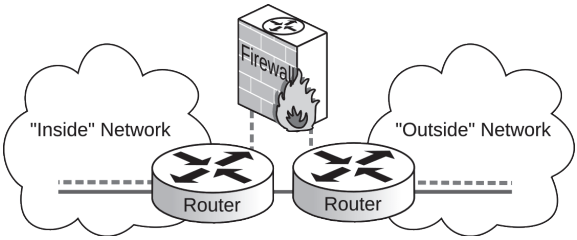


Fig. 3: Firewall bypass with two routers.

Fig. 3 shows a static firewall bypass where traffic is routed via two routers and a firewall which runs in routed mode, thus cannot be operated in transparent mode.

In this scenario, the rules for firewall bypassing can be expressed with the corresponding routes on each router. The routes have to be kept in sync in all routing tables which is non-trivial to automate.

Running a firewall in routed mode makes the firewall appear as a hop on Layer-3 and decreases the forwarded IP packets' time to live whereas in transparent mode, the firewall acts as a Layer-2 bridge device.

2.2 SDN and OpenFlow

SDN separates a network's data plane from its control plane. Unlike traditional standalone switches, SDN switches require a controller to configure their forwarding behaviour. A so-called southbound interface defines the communication between a switch and its controller.

OF is an SDN southbound interface. Its specification is published by the Open Networking Foundation (ONF) [Op15].

OF uses flow entries to define forwarding behaviour in a switch's OF tables. The flow entries are added and deleted on the switch by flow mod messages.

All published OF versions provide structures for flow entries that are sufficient to express firewall bypass rules. Flow match structures can match a 5-tuple plus the incoming switch port. A flow entry can hold instructions, e.g., an action can be applied to a packet. We need the output action which supports output switch port selection. Therefore, OF provides all capabilities needed for implementing an SDN-based firewall bypass.

Popular OF controller frameworks typically provide a programming environment with a complete OF feature set across different OF versions. This allows for flexible usage of the protocol but leads to large codebases. Those frameworks tend to use programming languages that come with heavyweight runtime environments which extend their codebase

even further and have to be regularly updated by system administrators. E.g., Ryu [Ry] is implemented in Python, Trema [Tr] uses Ruby, OpenDaylight [Op] and Floodlight [Fl] use Java. To avoid dependency on a heavyweight runtime environment, we decided not to use any of these popular frameworks. An additional concern with Ryu is its usage of the eventlet library that patches the Python standard library at runtime [Ev].

OFFWall is implemented from scratch in Rust [Ru]. Another Rust OF controller is rust_ofp which is not utilised because it is in an experimental development state. Its author elaborates on the advantages of an OF implementation in Rust rather than OCaml [Ba16].

2.3 Dynamic Firewall Bypass

A *dynamic* firewall bypass installs bypass rules traffic-driven on-demand. That means, all new flows are forwarded via the firewall. The controller may sample traffic leaving the firewall and select some flows for offloading. Thus, a dynamic firewall bypass dynamically generates the set of whitelisted flows and is more complex than a static firewall bypass.

Google holds a patent on a scalable stateful firewall design in OF-based networks [Pa14]. After processing new connections in a firewall, this design differentiates connections in flow state from connections in breach state. The connections in flow state are bypassed.

The authors of Science DMZ [Da13] suggest using a demilitarized zone (DMZ) for trusted network flows. They mention the possibility of using OF to implement the Science DMZ architecture.

SciPass [BR14] and NFShunt [MH15] build on Science DMZ and extend it with an OF-based firewall bypass. Both of them are dynamic firewall bypasses. SciPass provides support for an intrusion detection system in Science DMZ which also allows for dynamically identifying trustworthy network flows for the purpose of bypassing. NFShunt discusses the use of a Linux software firewall to deliver such information to an OF firewall bypass.

In previous work of ours [He17], static and dynamic firewall bypasses are compared. An OF-based dynamic firewall bypass with rule learning by sampling traffic via sFlow is suggested. The authors aim to use the bypass only in congestion situations. They present a proof of concept implementation and an analytical performance evaluation. Based on them they observed that this approach is only feasible if the OF switch can hold a large number of flow entries that exceeds the number of feasible flow entries of typical OF switches by an order of magnitude. This work is a follow-up and implements the simpler static firewall bypass.

3 Implementation of OFFWall

We describe OFFWall's requirements, the programming environment, the implementation design, and implementation details.

3.1 Requirements

We do not rely on an existing OF controller framework and implemented an OF controller from scratch that provides only the minimal set of OF features required to install the needed flow entries on the switch.

In our use case scenario there is no need for the controller to manage more than one switch or auxiliary connections. Therefore, OFFWall controls only a single switch. OFFWall preempts using any other OF controller if the OF switch cannot specify any additional controller.

This minimalistic approach aims at providing production-quality software that runs continuously for a long time. Furthermore, it should be possible to change the bypass rules without interrupting the execution of the switch or controller. The bypass rules should be validated semantically so that each rule contains an IP address belonging to the inside network. Any other checks are not involved. The operator has to pay attention to set only rules that describe trusted flows.

The target switch supports OF 1.0 and 1.3. Version 1.3 is implemented for the controller because it specifies the OF Extensible Match format which deprecates the 1.0 flow match structure and allows for more flexible matches.

3.2 Programming Environment

We decided to use the Rust programming language which is compiled to machine code like C and C++. It does not require a special runtime environment or a garbage collector. Rust has the additional benefits of guaranteed memory and thread safety, which eliminates some classes of run-time errors that are common causes of security problems. The guarantees are given at compile-time as opposed to many garbage-collected languages, which also provide memory safety at the expense of a runtime performance penalty. They come with a strong, static type system [MK14].

Version 1.0.0 of the implementation has 2626 lines of Rust code (excluding dependencies). 459 of these lines are type definitions constructed from parts of the *openflow.h* header file that is delivered with the OF specification [Op15]. That header file consists of 2335 lines of C code and defines the types needed for a complete OF implementation.

3.3 Implementation Design

Fig. 4 depicts an overview of the following implementation description.

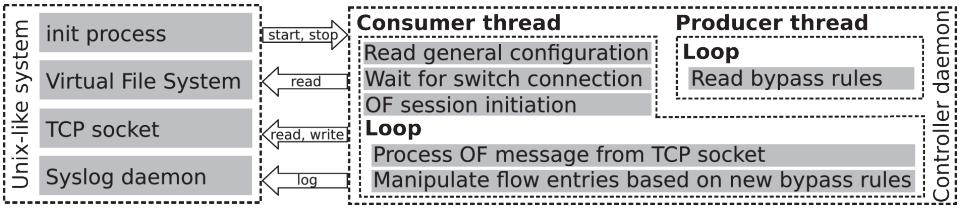


Fig. 4: Implementation overview and system interface usage of OFFWall.

The OF controller runs as a daemon on a Unix-like operating system. The daemon is started by the system's init process. It takes two command line arguments that are configuration file paths to a general configuration file and a bypass rules file. Sect. 3.4 explains them in detail.

The daemon uses two threads that communicate via a channel with a producer-consumer pattern. In the main thread (the consumer), the general configuration file is read and parsed. Then the process binds to a TCP socket and listens on it for an OF switch to connect as client. Sect. 3.5 explains the OF session initiation and the handling of the subsequent OF messages. If a connection error occurs, a new connection is offered immediately.

In the second thread (the producer), the bypass rules file is watched via the operating system's file watching mechanism. If no such mechanism is available, the file is polled once per second. When the file changes, this thread parses it and sends the modified 5-tuples to the main thread.

Various logging levels are implemented via the standard Unix logging facility Syslog and can be configured as an optional command line option.

3.4 Configuration Files

The general configuration file has INI syntax. Fig. 5 provides an example. The file contains, e.g., the controller's IP and port that it uses to open a TCP connection, the target OF table ID, and the OF port numbers in use, corresponding to the four switch ports in Fig. 1.

Fig. 6 provides an example for the bypass rules file. It has CSV syntax and holds the 5-tuples that belong to the firewall bypass rule set. All fields have wildcard support with a *, which means exclusion from the match. IP addresses are given in CIDR notation which allows for address block matches. We implemented only support for TCP and UDP over IPv4 because there is no need for firewall bypassing any other protocol at our campus network.

```
[Connection]
uri=tcp:192.0.2.1:6633

[Table]
id=0

[Networks]
inside=10.0.0.1/32

[Ports]
inside=10
fw_in=11
fw_out=12
outside=13
```

Fig. 5: An INI configuration file.

```
# src_ip;src_port; dst_ip;dst_port; proto
10.0.0.1/32; *;  10.0.0.2/32; 6060;  TCP
10.0.0.1/32; *;  10.0.0.2/32; 7070;  UDP
10.0.0.2/32; *;  10.0.0.1/32; 8080;  TCP
10.0.0.2/32; *;  10.0.0.1/32; 9090;  UDP
```

Fig. 6: A CSV file with bypass rules.

3.5 OpenFlow Message Handling

The following describes the handling of OF messages in the main thread. They are sent and received via exactly one TCP socket. All messages start with a fixed-length OF header that contains the OF version, message type, the message's length and a transaction identifier [Op15]. This structure allows for a straightforward message deserialisation.

Once a switch connects, the handshake consisting of the message types OFPT_HELLO, OFPT_FEATURES_REQUEST and OFPT_FEATURES_REPLY is exchanged. If the switch does not support OF 1.3, the connection is terminated.

By default, the switch will notify the controller on some events. Therefore, an OFPT_SET_ASYNC message is sent to unsubscribe from any switch notifications, which avoids implementing unnecessary message types.

Then the controller installs the default flow entries that forward packets to the firewall. It uses OFPT_FLOW_MOD messages for this task. The flow entries are derived from the [Ports] section of the general configuration file.

The connection is fully set up at this stage and the controller may install bypass rules. In an infinite loop new OF messages are read and deserialised.

When the controller receives a keep-alive message (OFPT_ECHO_REQUEST), it sends a reply (OFPT_ECHO_REPLY).

After a message is handled, the inter-thread channel is checked for any changed 5-tuples. According to the changes, several OFPT_FLOW_MOD messages are sent to add or delete flow entries that cause a firewall bypass. Each 5-tuple causes the creation of two flow entries:

inbound and outbound. The incoming switch port is derived from the IP range of the inside network given in the [Networks] section of the general configuration file. The switch port augments the 5-tuple in the flow entry's match structure.

4 Validation and Deployment

We present two different test environments that we used for a functional validation of OFFWall and describe the performed tests. After the successful test phase, we deployed OFFWall in an extended setup that is also presented.

4.1 Experiments for Validation

We implemented unit tests for the OF controller as base validation. Additionally, we performed system tests in a virtual environment and with a production-like setup. We describe the latter two in the following.

4.1.1 Virtualised Testbed

Fig. 7 shows the virtualised testbed which is realised with the Mininet [Mi] virtualisation environment running on a Linux system. The virtual bypass switch uses Open vSwitch with the proposed OF controller. This setup resembles the SDN-based firewall bypass in Fig. 2.

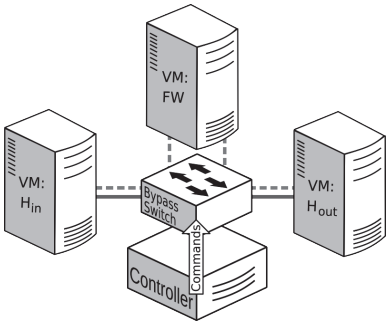


Fig. 7: Logical structure of the virtualised testbed.

The test network is set up as follows. An inside host VM (H_{in}) and an outside host VM (H_{out}) are both connected to the bypass switch enabling communication between them. To that end, they both listen on a UDP and on a TCP socket with their IP addresses and ports set according to Fig. 6. The firewall is represented by the VM FW with a link to the firewall-in port and another link to the firewall-out port of the bypass switch. We configure the VM FW such that it relays received frames between its two interfaces and logs every relayed packet.

The test verifies the correct installation, operation, and deinstallation of bypass rules on the bypass switch. We proceed in three steps: (1) transmission without bypass rules, (2) transmission with installed bypass rules, (3) transmission with deinstalled bypass rules. In each step we test bidirectional forwarding of TCP and UDP traffic between H_{in} and H_{out}. The logs of FW reveal whether traffic is bypassed.

In the first step, traffic is not bypassed and FW logs the communication. Then, the bypass rules from Fig. 6 are added to the controller's bypass rules file without restarting the bypass switch or the controller. The traffic is no longer logged by FW, i.e., bypass rules operate correctly. Finally, the bypass rules are removed from the bypass rules file without restarting the bypass switch or the controller. Now, FW logs the traffic again, i.e., bypass rules are successfully deinstalled.

4.1.2 Produktion-Like Physical Testbed

Fig. 8 depicts the production-like testbed. The bypass switch is an HP 5406zl and the firewall (FW) is a Cisco ASA 5500. The firewall is run in transparent mode and with logging enabled. It expects the incoming Ethernet frames to have a VLAN tag and also tags the outgoing frames. We did not implement VLAN support in OFFWall to keep the controller implementation minimal. As a workaround, we introduced an additional VLAN switch that VLAN-tags frames going to the firewall and untags VLAN frames coming from the firewall. This VLAN switch is a logical partition of the HP 5406zl so that no additional device is needed.

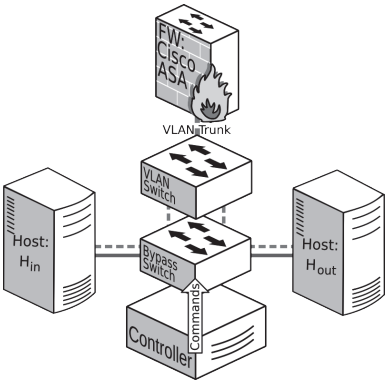


Fig. 8: Logical structure of the production-like testbed with VLAN tagging.

Each net is represented by a host (H_{in}, H_{out}) which are operated by arbitrary machines and operating systems. The tests are carried out as described for the virtualised testbed.

4.2 Deployment

After validating the OF controller implementation, OFFWall was deployed at the network of the Department of Computer Science of the University of Tuebingen. Fig. 9 depicts this setup that resembles the SDN-based firewall bypass in Fig. 2. The institutional network is the inside network whereas the backbone to the campus network and the Internet is the outside network.

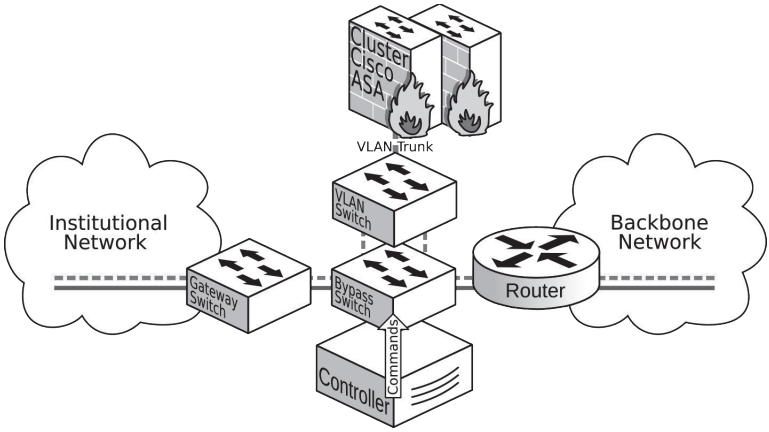


Fig. 9: OpenFlow-based firewall bypass with gateway switch and firewall cluster with VLAN tagging.

The OF-capable switch is HP 5406zl. The firewall consists of two Cisco ASA 5555 in cluster configuration and is run in transparent mode. The switch is partitioned like in the production-like testbed. One partition is responsible for the firewall bypass, the other is responsible for VLAN tagging and untagging.

When a conventional switch receives a frame from an unknown MAC address at one of its interfaces, it stores the MAC to interface mapping in its forwarding table. Future frames addressed to this MAC are then forwarded specifically to this interface (MAC learning). Usually, this learning function has to be implemented in OF switches, too. This occupies additional flow rules that are limited by OF switches. In order to avoid that, we install an HP Aruba 2930F gateway switch between the institutional network and the bypass switch. The gateway switch applies MAC learning and forwards internal and inbound traffic to the internal devices. As a consequence, the bypass switch does not need to apply MAC learning and we do not need to extend OFFWall.

5 Conclusion

As existing ACLs in switches are not flexible enough for implementation of a firewall bypass, we presented OFFWall, an OF controller with a reduced feature set for just this use case. It

is written in Rust and compiled to machine code with the purpose of runtime stability and without the need of a runtime environment.

We explained the implementation of OFFWall including configuration file structure. We reported tests on Mininet and a production-like hardware setup that were performed to validate OFFWall. We described the integration of OFFWall into the network of the Department of Computer Science of the University of Tuebingen which required an additional gateway switch for MAC learning.

References

[Ba16] Baxter, S.: OpenFlow 1.0 Protocol in Rust, 2016, URL: https://baxtersa.github.io/2016/12/30/rust-openflow-0x01.html.

[BR14] Balas, E.; Ragusa, A.: SciPass: A 100Gbps Capable Secure Science DMZ using OpenFlow and Bro, INDIS at SC '14, 2014, URL: https://scinet.supercomputing.org/workshop-sc14/?q=papers.

[Ci17] Cisco Systems, Inc.: CLI Book 1: Cisco ASA Series General Operations CLI Configuration Guide, 9.9, 2017.

[Da13] Dart, E.; Rotman, L.; Tierney, B.; Hester, M.; Zurawski, J.: The Science DMZ: A Network Design Pattern for Data-intensive Science. In: Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis. SC '13, 2013.

[Ev] Eventlet Contributors: Monkeypatching the Standard Library, URL: http://eventlet.net/doc/patching.html.

[Fl] Project Floodlight: Floodlight, URL: http://www.projectfloodlight.org/floodlight/.

[Ge] Germann, B.: OFFWall, URL: https://crates.io/crates/offwall.

[He17] Heimgärtner, F.; Schmidt, M.-T.; Morgenstern, D.; Menth, M.: A Software-Defined Firewall Bypass for Congestion Offloading. In: Proceedings of the International Conference on Network and Service Management. CNSM '17, 2017.

[MH15] Miteff, S.; Hazelhurst, S.: NFShunt: A Linux Firewall with OpenFlow-enabled Hardware Bypass. In: IEEE Conference on Network Function Virtualization and Software Defined Network. NFV-SDN '15, 2015.

[Mi] Mininet Team: Mininet, URL: http://mininet.org.

[MK14] Matsakis, N. D.; Klock II, F. S.: The Rust Language. In: Proceedings of the Conference on High Integrity Language Technology. HILT '14, 2014.

[Op] OpenDaylight Project, Inc.: OpenDaylight, URL: https://www.opendaylight.org.

[Op15] Open Networking Foundation: OpenFlow Switch Specification, Version 1.3.5, ONF, 2015.

[Pa14] Pani, A.: Scalable Stateful Firewall Design in OpenFlow based Networks, US8789135, Google Inc., 2014, URL: http://www.google.com/patents/US8789135.

[Ru] Rust Project: Rust, URL: https://www.rust-lang.org.

[Ry] Ryu SDN Framework Community: Ryu, URL: http://osrg.github.io/ryu/.

[Tr] Trema Project: Trema, URL: http://trema.github.io/trema/.

DNSSEC als Alternative zur klassischen CA

Daniel Feuchtinger, Helmut Reiser, Bernhard Schmidt[1]

Abstract: Der Betrieb klassischer Certificate-Authorities (CAs) wird organisatorisch immer aufwändiger. Die Komplexität, die Abhängigkeit von Browser-Herstellern durch eine Verankerung der Root-Zertifikate sowie diverse Sicherheitsvorfälle mit CAs und eine Vielzahl von „unbekannten" Root-Zertifikaten im Certificate-Store der Browser führen zu zunehmender Kritik an den etablierten Verfahren.

In dieser Arbeit wird untersucht, inwieweit DNSSEC mit seinen Mechanismen und Erweiterungen die Aufgaben einer klassischen CA übernehmen kann und welche Auswirkungen sich beim Einsatz solcher Zertifikate ergeben. Im Bereich der E-Mail Verschlüsselung lassen sich durch DNSSEC z.B. Zertifikate und Schlüssel sehr einfach verteilen, sind auch für Empfänger fremder Domains einfach zu nutzen und neben einer automatischen Verschlüsselung zwischen Mailservern ist auch eine vollautomatische Ende-zu-Ende Verschlüsselung möglich.

Keywords: DNSSEC; DANE; CA; PKI; Zertifikate

1 Einleitung

Die durch klassische Certificate-Authorities (CAs) bereitgestellte Public-Key-Infrastructure (PKI) hat viele technische und organisatorische Nachteile. Die immer länger werdenden und trotzdem unvollständigen Certificate-Revocation-Lists (CRLs), das Online-Certificate-Status-Protocol (OCSP) und die Pflege der Root-Zertifikate etwa in Browsern oder Betriebssystemen sollen hier als Beispiele für technische Probleme dienen. Die indifferente Verteilung von Vertrauen an eine Vielzahl von weitgehend unbekannten CAs und der damit fast zwangsläufig einhergehende Vertrauensbruch von Seiten einiger, nicht nur kleiner CAs [Fo11], sowie die Macht, die vor allem die Browser-Hersteller darüber haben, welchen CAs vertraut wird, sollen hier stellvertretend für die organisatorischen Probleme genannt werden. Das CA/Browserforum[2] definiert Richtlinien die von CAs zu erfüllen sind, damit deren Root-Zertifikate mit den Browsern ausgeliefert (verankert) werden. Die damit verbundenen organisatorischen und technischen Anforderungen haben sich in den letzten Jahren massiv verschärft. So wurden bspw. die Revalidierungszeiträume von 39 Monaten auf 825 Tage verkürzt. Die CA muss ihre Sicherheitsprozesse einmal jährlich nach den in [CA17] Abschnitt 17.1 angegebenen Standards (z.B. [ET13]) unabhängig auditieren lassen. Diese

[1] Leibniz-Rechenzentrum, Boltzmannstr. 1, 85748 Garching, daniel.feuchtinger@lrz.de, helmut.reiser@lrz.de, bernhard.schmidt@lrz.de

[2] https://cabforum.org

Regelungen können mit jeder Änderung der Guidelines verschärft werden und es droht der Verlust der Browserverankerung der Root-CAs bei Fehlern. Gleichzeitig versuchen einzelne Browser-Hersteller durch ihre Marktmacht, weitere Standards zu setzen oder Regelungen des CA/Browserforums in Frage zu stellen. Google implementiert in Chrome Certificate Transparancy [Ce17] und Mozilla stellt eine Auditierung nach dem ETSI Standard in Frage [Gr17]. Für Betreiber von CAs ergibt sich dadurch ein hohes Maß an Unsicherheit und das latente Risiko, dass Root-Zertifikate aus Browsern genommen werden und damit die Validierung für alle von dieser CA ausgestellten Zertifikate erheblich erschwert wird.

Das Domain-Name-Systen (DNS) ist ohnehin essentieller Bestandteil der klassischen PKI (OCSP baut auf DNS, Lets Encrypt validiert optional via DNS etc.). Durch DNSSEC werden die DNS-Informationen kryptographisch abgesichert und darauf aufbauend können mit DANE Zertifikate direkt kryptographisch sicher validiert werden. Auf den ersten Blick kann DNSSEC und DANE also eine klassische CA ersetzen. Welche Vorteile eine DNSSEC-DANE-PKI gegenüber den klassischen CAs hat und welche Voraussetzungen erfüllt sein müssen, um die klassische CA zu ersetzen, soll hier dargestellt werden.

2 Einführung DNSSEC und DANE

Im folgenden Abschnitt werden die Grundlagen für einen CA Einsatz von DNSSEC [GH10] und DANE [HS12] vorgestellt. Ist eine Domain DNSSEC-validiert, so ist jeder Teil des DNS-Baumes von der Wurzel bis zu dieser Domain kryptographisch mit Signaturen abgesichert. Die Baumstruktur des DNS legt die Zuständigkeiten für Signaturen und den Validierungspfad eindeutig fest. Es gibt nur eine „Root-CA“, die von der Internet Corporation for Assigned Names and Numbers (ICANN) verwaltete Root-Zone, alle weiteren CAs sind Sub-CAs mit eindeutig festgelegten Zuständigkeiten. Der öffentliche Schlüssel der Root-Zone ist der einzige Schlüssel, der initial verteilt werden muss. Das Beispiel lrz.de soll die Struktur einer DNSSEC-PKI verdeutlichen (Abbildung1). Für die Vertrauenskette ist neben der ICANN noch das Deutsche Network Information Center (DENIC), das die de-Zone verwaltet, und das LRZ, das die lrz.de-Zone und alle darunter liegenden Zonen verwaltet, zuständig. ICANN und DENIC haben nur die Aufgabe, die Vertrauenskette zum LRZ herzustellen, die Verwaltung der Zertifikate liegt organisatorisch ausschließlich beim LRZ. Technisch haben die Institutionen, die Teil der Kette sind (und nur die!), die Möglichkeit, diese Kette zu ändern und damit Zertifikate zu fälschen, das muss durch Verträge ausgeschlossen werden. Bei klassischen CAs haben alle Root-CAs (und ggf. auch Sub-CAs) die technische Möglichkeit, Zertifikate zu fälschen.

DNS-based Authentication of Named Entities (DANE) [DH15b; HS12] setzt auf DNSSEC [GH10] auf und verwendet DNS-Resource-Records, um Schlüsselinformationen zu speichern, z.B. vom Typ TLSA zur Veröffentlichung von Server-Zertifikaten oder vom Typ SMIMEA zur Veröffentlichung von S/MIME-Zertifikaten für die E-Mail-Verschlüsselung und Signatur (SMIMEA ist nicht Teil von DANE, sondern eine Ergänzung [HS17]).

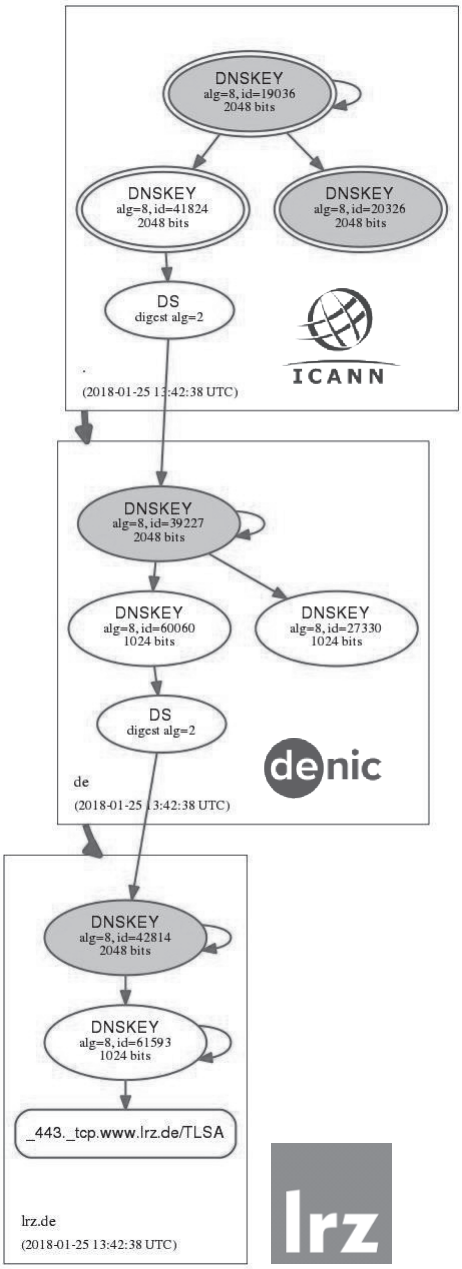


Abb. 1: DNSSEC/DANE Vertrauens-Struktur für den TLSA-Record zu www.lrz.de

Die dünnen Pfeile zeigen den technischen Validierungspfad mittels Signaturen und signierter Hashes, die dicken Pfeile zeigen die daraus folgende Delegation von Vertrauen auf Institutionsebene.

Ein Zertifikat kann durch Hinterlegen

0. des öffentlichen Schlüssels des Zertifikats im DER-Format oder
1. des Zertifikats im X.509 bzw. Public Key Infrastructure Exchange (PKIX) Format

(optional auch als sha256- oder sha512-Hash) im DNS in Form eines TLSA-Records referenziert werden. Die eigentliche Validierung erfolgt über eine der folgenden Varianten, die als Parameter (Usage Field) im TLSA-Record festgelegt wird (Nomenklatur siehe [Gu14]).

0. **CA Constraint (PKIX-TA)**: Der TLSA-Record gibt die vertrauenswürdige CA an, d.h. die CA, auf die das Zertifikat über die PKIX-Certification-Path-Validation zurückgeführt werden muss. Hier findet ein CA-Pinning durch DNSSEC/DANE statt.
1. **Service Certificate Constraint (PKIX-EE)**: Der TLSA-Record gibt das zu verwendende Zertifikat an, wobei zusätzlich die PKIX-Certification-Path-Validation durchgeführt werden muss. Hier findet ein Zertifikats-Pinning durch DNSSEC/DANE statt.
2. **Trust Anchor Assertion (DANE-TA)**: Das im TLSA-Record referenzierte Zertifikat dient als Trust-Anchor, d.h. ein Zertifikat ist vertrauenswürig, wenn es über eine Zertifikatskette auf den Trust-Anchor zurückführbar ist.
3. **Domain-Issued Certificate (DANE-EE)**: Dem über den TLSA-Record referenzierten Zertifikat wird vertraut, ohne weitere PKIX-Validierung. Damit kann auch die Prüfung der Gültigkeitsdauer und der Zertifikatssignatur entfallen, die Gültigkeitsdauer des Zertifikats wird durch die Gültigkeitsdauer des DNS-Records bzw. die der Signaturen festgelegt.

Es können mehrere TLSA-Records pro Domain angegeben werden. Um die Verwendung genauer zu spezifizieren, ist es vorgesehen, Protokoll und Port per DNS anzugeben: _port._protocol.domain. Um z.B. ein Zertifikat für den TCP-Port 443 des Hosts www.lrz.de anzugeben, wird folgender DNS-Knoten verwendet: _443._tcp.www.lrz.de.

Personenzertifikate sind nicht im DANE-RFC spezifiziert und mit den angesprochenen Mitteln ist es auch nicht möglich, eine Person im DNS abzubilden. Mit SMIMEA [HS17] ist ein Standard verfügbar, der einen DNS-Record spezifiziert, mit dessen Hilfe E-Mail-Adressen zu einer Domain referenziert werden können. Die Möglichkeiten, Zertifikate im DNS anzugeben werden vollständig von DANE übernommen, an die Stelle von _port._protocol in der Domain für den TLSA-Record kommt für den SMIMEA-Record der lokale Teil der E-Mail-Adresse als Hash (SHA256), ergänzt um `._smimecert`. Das Hashing löst einerseits Probleme der Zeichenkodierung (DNS ist hier wesentlich restriktiver als E-Mail), und kann andererseits

das Auslesen von Mail-Adressen via DNS verhindern. Ein Beispiel für einen SMIMEA-Record, der der E-Mail-Adresse hugh@test das Zertifikat des LRZ-Webservers zuordnet: `c93f1e400f26708f98cb19d936620da35eec8f72e57f9eec01c1afd6._smimecert.test. 86400 IN SMIMEA 3 1 1 71175DD9FE14CD41F1366BF5E82E25BCFFB46457BD7BF9AAEA37EFE443B0A449` Der SMIMEA-Record besagt, dass Mails von hugh@test mit dem im Record angegebenen Zertifikat signiert, und Mails an hugh@test mit dem öffentlichen Schlüssel aus dem Zertifikat verschlüsselt werden dürfen. Es können mehrere Zertifikate für eine E-Mail-Adresse via SMIMEA-Record angegeben werden. Ist die E-Mail-Adresse eindeutig einer Person zugeordnet (etwa durch eine Policy), so kann das im SMIMEA-Record referenzierte Zertifikat als Personenzertifikat betrachtet werden.

3 DNSSEC und DANE als PKI

DNSSEC und DANE stellen das Rüstzeug, um Zertifikate kryptographisch abgesichert einer Domain zuzuordnen. Das Ausstellen von Zertifikaten für eine Domain erfolgt durch das Anlegen und signieren von TLSA- oder SMIMEA-Records auf den für diese Domain autoritativen DNS-Servern. Beliebige Zertifikate können auf diesem Weg gleichzeitig signiert und veröffentlicht werden. Dabei ist zu beachten, dass die Bedeutung einiger Felder im Zertifikat ersetzt wird, wenn nicht TLSA-Records der Nutzungsvarianten PKIX-TA oder PKIX-EE (s. Abschnitt 2) verwendet werden. Diese erhalten die Bedeutung aller Zertifikatsfelder, da TLSA hier nur zusätzlich zur PKIX–Validierung gedacht ist. Bei den anderen Nutzungsvarianten ersetzen Informationen aus dem DNS einige Felder, die wichtigsten Beispiele sind:

- Die Gültigkeitsdauer des Zertifikats bestimmt sich durch die Gültigkeit der Signatur des TLSA-Records (RRSIG-Record).
- Der Common Name (CN) des Zertifikats wird durch die Domain des TLSA-Records ersetzt.
- Alternative Namen werden ignoriert, das Zertifikat kann aber auch über weitere TLSA-Records für weitere Domains verwendet werden.
- Die Signatur im Zertifikat wird gar nicht, teilweise oder vollständig geprüft, je nachdem, welche Nutzungsart der TLSA-Record definiert.
- Die Nutzungseinschränkung im Zertifikat wird ignoriert, eine Einschränkung ist durch _port._protocol gegeben.

Die DNSSEC/DANE-Validierung übernimmt ein DNSSEC-validierender DNS-Resolver und signalisiert dies über das AD-Flag in der DNS-Antwort. Die Kommunikation zwischen dem DNS-Resolver und dem Client ist nicht gegen Angriffe abgesichert, daher muss der Weg zwischen dem Resolver und dem Client sicher sein, zum Beispiel durch VPN, TSIG,

DNS-over-TLS oder am einfachsten durch die Nutzung eines lokalen validierenden Resolvers auf dem Client.

Ab Version 9.11 unterstützt der ISC BIND SMIMEA-Records, damit ist der produktive Einsatz auf einem authoritativen DNS-Server möglich. Resolver sind für unbekannte Resource Records im Allgemeinen transparent. Die Client-Unterstützung ist noch nicht so weit entwickelt, für Thunderbird gibt es das Great-DANE-Plugin[3] und für Mail-Server gibt es Filter, die aus- und eingehende Mails automatisch via SMIMEA-Record verschlüsseln, letzteres wird auch schon von Dienstleistern angeboten[4].

Ein Zertifikat ist aus DANE-Perspektive solange gültig, wie es per TLSA-Record validiert werden kann, ein Rückruf erfolgt durch das Löschen des TLSA-Records inklusive der dazugehörenden RRSIG-Records. Entscheidend für den Zeitpunkt der Wirksamkeit des Rückrufs ist die Gültigkeitsdauer der Signaturen des TLSA-Records, und der DNS-Records, die bei der Validierung verwendet werden (das sind mindestens noch die Delegations-Records). Nach Ablauf der Gültigkeit aller TLSA-Signaturen (durch wiederkehrendes Signieren der DNS-Zonen können mehrere Signaturen im Umlauf sein) kann dieses Zertifikat nicht mehr verwendet werden.

Ist ein DNSKEY kompromittiert und kann ein Angreifer beliebige Zertifikate für die betroffenen Domain inklusive Subdomains „ausgeben“, so ist das mit der Kompromittierung eines privaten CA-Schlüssels bei PKIX vergleichbar. Alle RRSIG-Records (d.h. die DNSSEC-Signaturen, auf denen das Vertrauen in die Zertifikate beruht) sind nicht mehr vertrauenswürdig und ein DNSSEC-Key-Rollover muss durchgeführt werden. Die RRSIG-Records zu den TLSA-Records, d.h. auch möglicherweise gefälschte Zertifikate, sind spätestens nach einem abgeschlossenen Key-Rollover wirksam zurückgerufen. Daraus ergibt sich ein Vorteil gegenüber den klassichen CAs, vorausgesetzt, man nutzt TLSA-Records nicht als Trust-Anchor (PKIX-TA oder DANE-TA): Bei der klassischen CA müssen alle ausgegebenen Zertifikate ausgetauscht werden, wenn der Signaturschlüssel kompromittiert wurde, bei DNSSEC/DANE genügt ein Key-Rollover. Die vor der Kompromittierung gültigen Zertifikate bzw. TLSA-Records können alle mit neuen DNSSEC-Signaturen weiter verwendet werden.

DNSSEC sieht nicht vor, widerrufene Zertifkate zu speichern, es gibt also keine wachsende Liste an zurückgerufenen und abgelaufenen Zertifikaten die gepflegt werden muss. Will man ein neues Zertifikat bzw. Schlüsselpaar verwenden, das alte aber noch für die Validierung von Signaturen behalten, so kann man das alte in Form eines Hashes verfügbar lassen und nur das neue in Form eines vollständigen Zertifikats oder öffentlichen Schlüssels zur Verfügung stellen. Kompromittierte Schlüssel und die dazugehörenden Zertifikate werden vollständig aus dem DNS gelöscht und sind damit im Nachhinein nicht mehr zuzuordnen.

Klassische CAs bieten die Möglichkeit, Signaturen mit zurückgerufenen Zertifikaten zu

[3] https://addons.mozilla.org/de/thunderbird/addon/great-dane-smim/e

[4] https://www.tb-itf.de/faq-eintrag/automatische-verschluesselung-mit-smimea.html

validieren, allerdings stellt sich hier die Frage nach dem Sinn, da es keine Möglichkeit gibt, das Mindestalter einer Signatur nachzuweisen. Es ist also nicht möglich festzustellen, dass eine Signatur vor der Kompromittierung des Schlüssels erstellt wurde und damit ist die Signatur wertlos geworden.

Wird trotzdem eine Zertifikats-History in Form der zurückgerufenen Zertifikate benötigt, so müssen eigene Lösungen gefunden werden.

4 Gegenüberstellung von DNSSEC/DANE und klassischen CAs

Für den Vergleich von DNSSEC/DANE und klassischen CAs werden hier einige der jeweils interessanten Voraussetzungen umrissen und anschließend Vor- und Nachteile diskutiert. Die kryptographischen Details werden nicht angesprochen, da sich DNSSEC und die klassischen CAs weitgehend auf die selben Algorithmen stützen.

Um ein Zertifikat gegen die klassischen CAs validieren zu können, müssen die Root-Zertifikate der CAs lokal verfügbar und aktuell sein. Zurückgerufene Zertifikate müssen gepflegt werden, beispielsweise in Form von Certificate-Revocation-Lists (CRLs), die mit der Zeit immer weiter anwachsen können und jederzeit verfügbar sein müssen. Die daraus resultierenden Probleme sollte das Online-Certificate-Status-Protocol (OCSP) lösen, das den Zertifikatsstatus dynamisch abfragbar macht. Dazu muss ein Server, ein sog. OCSP-Responder, der über widerrufene Zertifikate Auskunft gibt, dauerhaft und überall verfügbar sein. Letzteres ist schon ohne potentielle Angreifer ein schwer zu lösendes Problem. CRL-Verteilungspunkte und OCSP-Responder werden in den Zertifikaten als URL angegeben und hängen damit beide vom DNS ab. OCSP ist also über Routing und DNS angreifbar, die Anfragen des Clients können umgelenkt werden. Da OCSP ein „try-again-later“ vorsieht, welches nicht signiert werden muss [Ma09; Sa13] und welches client-seitig oft nicht zu einem Fehler führt, kann OCSP über einen Man-in-the-Middle-Angriff vollständig ausgehebelt werden. OCSP kann auch für Tracking missbraucht werden, da für jede TLS-Verbindung eine Verbindung zum OCSP-Responder aufgebaut werden muss. Aufgrund dieser Schwächen führt Chrome standardmäßig keine OCSP-Abfragen durch sondern verwendet vom Chrome-Projekt gepflegte(!) CRL-Sets, die sporadisch über Software-Updates installiert werden müssen.

Die Verwendung von DNSSEC/DANE als CA erfordert die Validierung und damit auf Client-Seite einen DNSSEC-validierenden DNS-Resolver. Insbesondere für mobile Geräte ist ein validierender DNS-Resolver in „sicherem Abstand“ nicht immer verfügbar, daher ist die Integration eines validierenden Resolvers im Betriebssystem eine Voraussetzung für eine breite Anwendung von DNSSEC-DANE als CA. Für Server-Anwendungen ist ein lokaler validierender DNS-Resolver kein Problem, daher ist der Einsatz einer DNSSEC/DANE-CA hier einfacher. Client-Anwendungen müssen die Verwendung von DANE unterstützen. Während die Unterstützung PKIX weit verbreitet ist, wird DANE in größerem Umfang derzeit nur für die Kommunikation von SMTP-Servern untereinander

verwendet. Für Firefox und Chrome gibt es ein DANE-Plugin, das aber die Validierung über die klassischen CAs nicht ersetzt. Scheitert die PKIX-Certification-Path-Validation, so wird vom Browser die übliche Warnung angezeigt, auch wenn die Validierung über das DANE-Plugin erfolgreich war. Eine DNSSEC/DANE-CA kann jeder selbst betreiben und die Sicherheitsanforderungen für die eigene Zone definieren und überwachen. Die Betreiber der Root- und der Second-Level-Domain-Zonen sind allerdings festgelegt.

Klassische CAs	
Vorteile	**Nachteile**
• Breite Unterstützung durch Betriebssysteme und Browser • Unabhängigkeit von sicherem Netz	• Root-CA-Zertifikate müssen auf allen Endgeräten gepflegt werden • Zertifikatsrückruf ist angreifbar • Vertrauen breit gestreut (Beispiel Firefox: über 150 Root-CA-Zertifikate, über 60 CAs[5] • Hohe technische Komplexität • Mißbrauchsfälle • Hoher, und weiter steigender Akkreditierungsaufwand • Abhängigkeit von Browser- und Betriebssystemherstellern • Kosten • Zuständigkeit unklar (auch mit CAA-Records!) • Kompromittierung des Signierschlüssels erfordert Austausch aller Zertifikate

DNSSEC/DANE	
Vorteile	**Nachteile**
• Baut auf vorhandene Infrastruktur auf • Zertifikatsrückruf funktioniert • Geringe Kosten, wenn DNSSEC schon vorhanden ist • Eigene Domains können flexibel und unabhängig selbst, oder von Dienstleistern (ohne Akkreditierung) verwaltet werden • Unabhängigkeit von sicherem Netz (mit lokalem Resolver) • Zuständigkeit klar • Kompromittierung des Signierschlüssels erfordert keinen Austausch von Zertifikaten	• Unterstützung durch Anwendungen fehlt • Zeitverzögerter Widerruf • Zurückgerufene und abgelaufene Zertifikate sind nicht über DNS verfügbar • Abhängigkeit von den Verwaltern der Root-Zone und der Top-Level-Domain-Zonen • Vertrauenswürdige validierende DNS-Resolver sind nicht überall verfügbar

[5] https://wiki.mozilla.org/CA

Eine private CA ist auch mit DNSSEC/DANE möglich, hat aber ähnliche Nachteile, wie eine private klassische CA, bei der das Root-Zertifikat auf den relevanten Clients installiert werden muss. Für eine private DNSSEC/DANE-CA muss ein validierender DNS-Resolver mit DNSSEC-Lookaside-Validation verwendet werden, der für eine bestimmte Zone einen eigenen DNSSEC-Trust-Anchor definiert.

5 Anwendungsfall E-Mail

Die Möglichkeiten der DNSSEC/DANE-CA werden hier für die Signatur und Verschlüsselung von E-Mails erörtert. Die Verwendung von DANE für SMTP-Server-Kommunikation ist bereits etabliert und in einem eigenen RFC [DH15a] beschrieben.

DNS mit DNSSEC gesicherten SMIMEA-Records kann die Funktion einer Zertifikatsdatenbank übernehmen und erlaubt das (auch automatisierte) Nachschlagen von Zertifikaten. Die Validierung eines Zertifikats kann eine klassische CA übernehmen, sie kann aber auch ausschließlich durch DNSSEC/DANE und unabhängig von einer klassischen CA erfolgen (SMIMEA-Nutzung DANE-TA und DANE-EE).

Auswertung des SMIMEA-Records durch Mail-Server: Eine vergleichsweise einfach umzusetzende Variante ist der Einsatz auf Mail-Servern, da ein lokaler validierender DNS-Resolver einfach eingerichtet, und nur auf den Mail-Servern notwendig ist. Der SMTP-Server kann so konfiguriert werden, dass jede ausgehende Mail verschlüsselt wird, wenn ein DNSSEC-gesicherter SMIMEA-Record für den Empfänger verfügbar ist. Für jede eingehende Mail kann genauso verfahren werden. Die Mails liegen dann automatisch verschlüsselt auf dem Server und können erst durch den Empfänger entschlüsselt werden. Passiert das auf dem Mail-Submission-Server, so geht die Mail nicht im Klartext über das Netz (vorausgesetzt, der Mail-Client verwendet TLS für die Verbindung zum Mail-Submission-Server) und wird online nur verschlüsselt gespeichert. Der SMIMEA-Record bietet keine Möglichkeit, die Verwendung zu differenzieren, also z.B. ein Flag „immer verschlüsseln“, oder „nur für Signaturverifikation“ zu setzen. Über die Art des Records lässt sich das aber zumindest teilweise erreichen, etwa indem nur ein Fingerprint hinterlegt wird, mit welchem zwar nicht automatisch verschlüsselt, aber eine Signatur überprüft werden kann. Die Wahlmöglichkeiten könnte man auch für den Nutzer konfigurierbar machen, sowohl beim Versenden, d.h. auf dem Mail-Submission-Server, als auch für Postfächer, d.h. für alle eingehenden Mails. Mailfilter für diese Szenarien sind bereits im Einsatz[6][7].

Auswertung des SMIMEA-Records durch Mail-Clients: Die Unterstützung von SMIMEA durch Mail-Clients ist noch weniger ausgereift, als die für Mail-Server. Auch hier

[6] https://www.tb-itf.de/faq-eintrag/automatische-verschluesselung-mit-smimea.html

[7] https://www.heise.de/newsticker/meldung/DANE-Automatische-Mail-Verschluesselung-mit-S-MIME-3041530.html

ist ein Problem die Verfügbarkeit eines vertrauenswürdigen validierenden DNS-Resolvers. Für Thunderbird gibt es ein vielversprechendes Plugin Great DANE[8], das die Möglichkeit bietet, Zertifikate für Empfänger zur Verschlüsselung, und für Absender zur Verifikation der Signatur automatisch via SMIMEA-Records zu finden. Die Great-DANE-Engine[9] ist auch unabhängig von Thunderbird nutzbar und kann z.B. in Webmail-Software integriert werden.

DNSSEC/SMIMEA-CA: Aus Sicht eines CA-Betreibers können SMIMEA-Records auf drei Arten eingesetzt werden:

1. Aufbauend auf eine klassische CA, jedes ausgegebene Zertifikat kann (ggf. auf Wunsch) als SMIMEA-Record veröffentlicht werden (PKIX-EE, ggf. zusätzlich PKIX-TA).
2. Aufbauend auf eine Self-Signed-CA, die SMIMEA-Records dienen hier mit DNSSEC als kryptographische Absicherung. (DANE-EE, ggf. zusätzlich DANE-TA)
3. Der Nutzer kann ein beliebiges Zertifikat vorlegen und über einen SMIMEA-Record veröffentlichen lassen (DANE-EE).

In den ersten beiden Varianten muss der Nutzer ein Certificate-Signing-Request (CSR) einreichen und z.B. persönlich seine Identität bestätigen, bevor ein Zertifikat ausgegeben und im DNS veröffentlicht wird. Mit der zweiten Variante hat man die Möglichkeit, die Art der per DNS ausgegebenen Zertifikate zu kontrollieren und den Einsatz zu protokollieren, analog zur klassischen CA (Zeitpunkt der Ausgabe, Zeitpunkt der Löschung aus dem DNS), auch wenn für den Einsatz (zumindest bei DANE-EE) eine Signatur durch die Self-Signed-CA überflüssig ist, da die Authentizität des Zertifikats ausschließlich durch DNSSEC verifiziert wird. Bei der dritten Variante ist es möglicherweise sinnvoll, über eine Policy einige Anforderungen an ein Zertifikat zu formulieren, z.B. um den Inhalt der (nicht verwendeten) Zertifikatsfelder unauffällig zu halten. Wenn diese Anforderungen automatisiert geprüft werden können, ist es sogar möglich, dass jeder Nutzer sein Zertifikat selbst hochlädt, nachdem er sich authentisiert hat.

6 Anwendungsfall Webserver-Zertifikat

Die Möglichkeiten von DNSSEC/DANE für Webserver-Zertifikate sind analog zur DNSSEC/SMIMEA-CA (vgl. Punkt 1 bis 3 des vorherigen Abschnitts). Da die verbreiteten Webbrowser DNSSEC/DANE nicht unterstützen und das Plugin für Firefox und Chrome die klassische PKIX-Verifikation nicht ersetzt, sondern nur ergänzt, ist der Nutzen einer DNSSEC/DANE-CA geringer, als bei DNSSEC/SMIMEA. DNSSEC/DANE bietet hier

[8] https://addons.mozilla.org/de/thunderbird/addon/great-dane-smime/
[9] https://github.com/grierforensics/Great-DANE-Engine

bisher „nur“ eine zusätzliche Sicherheit, allerdings auch keine Nachteile (bis auf die Pflege der TLSA-Records). Somit ist nur die erste Variante für Webserver geeignet, die für eine breitere Öffentlichkeit gedacht sind, Varianten 2 und 3 können nur als Spezialfälle dienen, wo bei den Nutzern der Seite das DANE-Plugin, ein validierender DNS-Resolver, und entsprechendes Wissen vorausgesetzt werden können (der Browser wird eine Seite ohne PKIX-Certification-Path-Validation als unsicher markieren und eine Warnung ausgeben, auch mit DANE-Plugin), was eine breite Anwendung in dieser Form verhindert. Abhilfe könnten natürlich die Browserhersteller durch Integration von DNSSEC/DANE in die Browser schaffen, oder durch ein Plugin, das die klassische PKIX-Validierung ersetzen kann.

7 Zusammenfassung

DNS-Daten sind essentiell für alle Internetanwendungen, der Zugriff auf das DNS muss daher hoch verfügbar sein. In vielen Fällen, nicht zuletzt für klassische CAs, sind DNS-Daten von sicherheitsrelevanter Bedeutung und schützenswert. Die Verbreitung von DNSSEC wird weiter zunehmen und die Datenbankfunktion des DNS für sicherheitsrelevante Anwendungen wie die Verteilung öffentlicher Schlüssel und Zertifikate verfügbar machen, während die Validierung der Zertifikate noch eine klassische CA übernimmt.

Mit der bereits vorhandenen Zuordnung von Domains zu Inhabern ist eine Aufgabe einer CA erfüllt: Der Inhaber der Domain kann die Verantwortung für Zertifikate und Schlüssel zu dieser Domain übernehmen bzw. an Dritte (z.B. den Betreiber einer klassischen CA) delegieren. In einem weiteren Schritt kann DNSSEC/DANE dann die Validierung von Zertifikaten übernehmen und bildet damit eine hochverfügbare, einfache und elegante CA, die mit nur einem Root-Zertifikat auskommt und mit einer weltweit verfügbaren Datenbank für Schlüssel und Zertifikate besticht. Dafür ist allerdings noch eine weitere Verbreitung von DNSSEC und validierenden Resolvern nötig. Ein entscheidender Schub könnte etwa durch die Unterstützung der Browser, oder der Smartphonebetriebssysteme kommen. Letzere bringen jetzt schon oft einen eigenen DNS-Resolver mit. Könnte dieser DNSSEC validieren, möglicherweise sogar selektiv (je nach Anforderung einer App), so würde das die DNSSEC/DANE-CA einem großen Anwendungskreis verfügbar, und die Ersetzung der klassischen CA realistisch machen. In kontrollierten Umgebungen wird DNSSEC/DANE bereits produktiv eingesetzt, wie z.B. in einem bayernweiten Projekt [Du17] zur sicheren Mail-Server-Kommunikation nach RFC7672 [DH15a]. Ein weiteres Anwendungsfeld könnte die automatische Verschlüsselung von Nutzer-Emails durch Mail-Submission-Server sein.

Literatur

[CA17] CA/Browser Forum: Guidelines For The Issuance And Management Of Extended Validation Certificates, Techn. Ber. Version 1.6.6, CA/Browser Forum, 2017, URL: https://cabforum.org/wp-content/uploads/EV-V1_6_6.pdf.

[Ce17] Certificate Transparency: What is Certificate Transparency?, Techn. Ber., Google, 2017, URL: https://www.certificate-transparency.org/what-is-ct.

[DH15a] Dukhovni, V.; Hardaker, W.: SMTP Security via Opportunistic DNS-Based Authentication of Named Entities (DANE) Transport Layer Security (TLS), RFC 7672, RFC Editor, Okt. 2015.

[DH15b] Dukhovni, V.; Hardaker, W.: The DNS-Based Authentication of Named Entities (DANE) Protocol: Updates and Operational Guidance, RFC 7671, RFC Editor, Okt. 2015.

[Du17] Duscha, S.; Schmidt, B.; Feuchtinger, D.; Reiser, H.: Einführung von DNSSEC und DANE im Bayerischen Hochschulnetz. In (Eibl, M.; Gaedke, M., Hrsg.): INFORMATIK 2017. Gesellschaft für Informatik, Bonn, S. 763–772, 2017.

[ET13] ETSI: Electronic Signatures and Infrastructures (ESI); Policy requirements for certification authorities issuing public key certificates, Techn. Ber. ETSI 102 042 V 2.4.1, European Telecommunications Standards Institute (ETSI), 2013.

[Fo11] Fox-IT), J. P. (: Interim Report, DigiNotar Certificate Authority breach, "Operation Black Tulip", Techn. Ber., FOX-IT, 2011, URL: https://cryptome.org/0005/diginotar-insec.pdf.

[GH10] Gould, J.; Hollenbeck, S.: Domain Name System (DNS) Security Extensions Mapping for the Extensible Provisioning Protocol (EPP), RFC 5910, RFC Editor, Mai 2010.

[Gr17] mozilla.dev.security.policy Group: ETSI audits not listing audit periods, Techn. Ber., Google Groups, 2017, URL: https://groups.google.com/forum/#!msg/mozilla.dev.security.policy/zAEoGqdtl6E/Q_Fr41V3BAAJ.

[Gu14] Gudmundsson, O.: Adding Acronyms to Simplify Conversations about DNS-Based Authentication of Named Entities (DANE), RFC 7218, RFC Editor, Apr. 2014.

[HS12] Hoffman, P.; Schlyter, J.: The DNS-Based Authentication of Named Entities (DANE) Transport Layer Security (TLS) Protocol: TLSA, RFC 6698, http://www.rfc-editor.org/rfc/rfc6698.txt, RFC Editor, Aug. 2012, URL: http://www.rfc-editor.org/rfc/rfc6698.txt.

[HS17] Hoffman, P.; Schlyter, J.: Using Secure DNS to Associate Certificates with Domain Names for S/MIME, RFC 8162, RFC Editor, Mai 2017.

[Ma09] Marlinspike, M.: Defeating OCSP With The Chara ter '3', Techn. Ber., 2009, URL: http://www.thoughtcrime.org/papers/ocsp-attack.pdf.

[Sa13] Santesson, S.; Myers, M.; Ankney, R.; Malpani, A.; Galperin, S.; Adams, C.: X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP, RFC 6960, http://www.rfc-editor.org/rfc/rfc6960.txt, RFC Editor, Juni 2013, URL: http://www.rfc-editor.org/rfc/rfc6960.txt.

Müller, Neumair, Reiser, Dreo Rodosek (Hrsg.): 11. DFN-Forum Kommunikationstechnologien,
Lecture Notes in Informatics (LNI), Gesellschaft für Informatik, Bonn 2018 69

Realisierung von sicheren Over-the-Air Updates für ESP8266-basierte IoT-Endgeräte

Dustin Frisch[1] Sven Reißmann[2] Christian Pape [1] Sebastian Rieger[1]

Abstract: Nicht zuletzt durch Trends wie Smart Home und Industrie 4.0 hat die Zahl der Endgeräte im Bereich Internet of Things (IoT) in den vergangenen Jahren stark zugenommen. Häufig werden diese Endgeräte als kostengünstige kleine eingebettete Systeme in großen Stückzahlen ausgerollt und über viele Jahre verwendet. Sie stellen nicht nur bedingt durch ihre knappen Ressourcen und eingeschränkten Schutzmechanismen ein Datenschutzrisiko für ihre Betreiber dar, sondern bilden aufgrund ihrer großen Verteilung und Angriffsfläche auch ein Sicherheitsrisiko für das gesamte Internet. Um diesem Risiko entgegenzuwirken, müssen Sicherheitsupdates für IoT-Endgeräte regelmäßig und zeitnah verteilt werden. Dies verdeutlichen auch die in den letzten Jahren aufgetretenen Angriffe, die auf IoT-Endgeräten aufgesetzt haben (z.B. Mirai Botnet). Das vorliegende Paper beschreibt ein nachhaltiges und stabiles Verfahren für die Bereitstellung kryptographisch gesicherter Over-the-Air Firmware-Updates für eingebettete Systeme basierend auf dem verbreiteten ESP8266 Mikrocontroller. Im Gegensatz zu anderen Over-the-Air-Verfahren werden der unterbrechungsfreie Betrieb des Mikrocontrollers und fehlertolerante Updates realisiert. Da der ESP8266 aufgrund seiner knappen Ressourcen keine vollständige HTTPS- bzw. TLS-Unterstützung bietet, wurde eine separate kryptographische Überprüfung bereitgestellter Updates implementiert. Das vorgestellte Verfahren umfasst den Build-Prozess des Updates, dessen automatische digitale Signatur sowie die Verteilung und Installation auf den Endgeräten. Darüber hinaus wird die Verwendung des Verfahrens zur Abwehr von auf IoT-Endgeräten basierenden Angriffen aufgezeigt.

Keywords: Internet of Things; Secure Remote Firmware Updates; Over-the-Air Updates; ESP8266

1 Einleitung

Das Internet of Things (IoT) hat zu einem großen Wachstum von sogenannten Smart Devices geführt, die z.B. Sensoren und Aktoren über das Internet miteinander verbinden. Bestehende Geräte wie z.B. Türschlösser, Lampen, Waschmaschinen etc. werden hierbei um smarte Funktionen erweitert, um sie z.B. aus der Ferne zu steuern und zu überwachen. Um diese smarten Funktionen zu implementieren, werden kleine eingebettete Systeme in die Geräte eingebaut, die z.B. die Anbindung an ein drahtloses Netz ermöglichen. Implementiert werden die Funktionen als Software, bzw. als Firmware, auf den eingebetteten Systemen. Die Firmware dient dabei zum Einen der Anbindung spezifischer Hardware (vgl. Sensoren und Aktoren) und bietet zum Anderen allgemeine Funktionen, vergleichbar der Aufgabe eines

[1] Hochschule Fulda, Angewandte Informatik, [vorname.nachname]@informatik.hs-fulda.de
[2] Hochschule Fulda, Rechenzentrum, sven.reissmann@rz.hs-fulda.de

Betriebssystems, unabhängig vom konkreten Einsatzgebiet (z.B. Anbindung an WLAN-Netze etc.). Anforderungen und Funktionen von eingebetteten Systemen werden häufig als unveränderlich angenommen. Allerdings zeigt sich beim Einsatz der Systeme in der Realität, dass sich allein durch Anforderungen der Umgebung in der sie betrieben werden, häufig doch Veränderungen ergeben, an die die Systeme angepasst werden müssen. Beispiele hierfür sind Veränderungen oder Erweiterung des Verhaltens der Systeme, Anpassungen der Netzanbindung (vgl. Reaktion auf neue Angriffe etc.) oder Fehlerbehebung bzw. insb. Sicherheitsupdates im Laufe des Betriebs. In den meisten Fällen können diese Anpassungen allein durch die Änderung der Firmware erfolgen, während die Hardware unverändert bleibt. Um die Firmware zu aktualisieren, müssen die Systeme eine geeignete Schnittstelle bieten, die in der Regel auch die Konfiguration des Systems sowie Informationen und Checks zur aktuellen Firmware verwaltet. In aller Regel erfordern die Schnittstellen für das Firmware-Update einen direkten physischen Zugriff auf das System. Insbesondere bei einer großen Anzahl und räumlichen Verteilung von Endgeräten sind Firmware-Updates, die einen direkten Zugriff auf das System erfordern, jedoch kaum realisierbar. Abhilfe bieten Schnittstellen, die ein *Over-the-Air (OTA)* Update aus der Ferne ermöglichen, solange das Endgerät über ein Netz erreicht werden kann. Das Update kann in diesem Fall ohne direkten Zugriff über die reguläre Netzwertschnittstelle erfolgen, ohne eine spezielle Schnittstelle zu erfordern. *OTA*-Updates ermöglichen ein automatisiertes Ausrollen von neuen Firmware-Versionen und Patches auf eine große Anzahl verteilter Endgeräte. Die Automatisierung kann dabei im Sinne einer Continuous Delivery Tests und Prozesse für die fehlertolerante Auslieferung der Firmware beinhalten. Hierbei können Updates z.B. zunächst automatisiert auf Testgeräten eingespielt und überprüft werden. Sicherheitsupdates können priorisiert im laufenden Betrieb ausgerollt werden, während Funktionsupdates verzögert verteilt werden, z.B. sobald das Endgerät zeitweise nicht verwendet wird. Über das Netz kann zusätzlich ein Monitoring des Endgeräts und der Firmware-Updates erfolgen, um sicherzustellen, dass alle Updates erfolgreich waren und sich das Endgerät im gewünschten Zustand befindet.

Die verbleibenden Abschnitte des Papers gliedern sich wie folgt. Abschnitt 2 nennt verwandte Arbeiten. In Abschnitt 3 werden die Anforderungen an das im nachfolgenden Abschnitt 4 erstellte Konzept für einen sicheren und robusten *OTA*-Update-Mechanismus definiert. Eine Referenzimplementierung folgt im Abschnitt 5. Abschließend zieht Abschnitt 6 ein Fazit und gibt einen Ausblick auf weitere Arbeiten.

2 Verwandte Arbeiten

Drahtlose Netze für Sensoren und Aktoren sind eine essentielle Grundlage für die Realisierung von Industrie 4.0 Infrastrukturen und modere Fertigungs- und Lieferprozesse. Günstige Programmable Logic Controller (PLC) und Cloud Computing ermöglichen und treiben diese neuen Paradigmen im Produktionsbereich gleichermaßen [NS16]. In [Di15] wird ein multifunktionales, kostengünstiges und drahtloses Sensornetz unter Verwendung des *ESP8266* PLCs vorgestellt. Die Verwendung eines *ESP8266* PLCs in Kombination mit

einem Raspberry Pi als Basisstation wird in [Th16] gezeigt. Der Artikel [KS16] präsentiert eine Heimautomatisierungslösung basierend auf einer *MQTT* Message Queue mit *ESP8266*-basierenden Sensoren und Aktoren. Die Kontrolle von smarten Lampen mit PLCs wird in [WKM16] zusammengefasst. Firmware-Update-Mechanismen werden in diesen Veröffentlichungen leider nicht adressiert. Die Bedeutung regelmäßiger Sicherheitsupdates für aktuelle IT-Infrastrukturen wurde in [Be16] zusammengestellt. Ein Ansatz für dezentrale Software Updates in Contiki-basierten IoT-Umgebungen wurde in [Ru16] vorgestellt. In [We16] werden Software Updates für Java-basierte Endgeräte präsentiert. Beide Lösungen sind für kleine Mikrocontroller mit geringen Ressourcen nicht anwendbar. [MFE12] beschreibt ein Diagnose- und Update-System für Embedded Software von Electronic Control Units in Fahrzeugen. Die Verwendung sicherer Firmware-Updates in der Automobilindustrie wird in [NL08] behandelt. In [Go16] wird ein Konzept für *Over-the-Air* Updates für *ESP8266* PLCs beschrieben. Die Updates sind allerdings nicht fehlertolerant im laufenden Betrieb möglich und erfordern ein separates Gateway.

3 Anforderungen

Sobald ein neues Firmware-Release vorliegt, soll das Update automatisch und ohne manuelle Interaktion auf die Endgeräte ausgerollt werden. Endgeräte sollen für sie passende neue Firmware-Versionen erkennen, herunterladen, automatisch installieren und, sofern bei der Installation kein Fehler auftritt, die neue Firmware durch einen Neustart anwenden. Um einen minimalen Wartungsaufwand zu erzielen, soll der Update-Prozess möglichst robust und fehlertolerant sein. Sofern während einem Update ein Fehler auftritt, soll das Endgerät weiterhin fehlerfrei funktionieren. Firmware-Updates müssen über die WLAN-Verbindung während des regulären Betrieb möglich sein ohne die Funktion des Geräts zu beeinträchtigen. Der Update-Prozess muss über nicht vertrauenswürdige WLAN-Netze und das Internet sicher möglich sein. Um zu verhindern, dass Angreifer manipulierte Firmware-Versionen einspielen können, soll eine digitale Signatur der Updates realisiert und überprüft werden. Der Update-Prozess soll nur dann angestoßen werden, wenn neue Firmware-Versionen vorliegen, und möglichst schnell ablaufen, um den unterbrechungsfreien Betrieb des Endgeräts zu sichern. Zur Erleichterung der Verwaltung und Überwachung von Endgeräten sollen für den Update-Prozess relevante Informationen (z.B. installierte Firmware-Version) von den Endgeräten abrufbar sein. Bei einem Update muss sichergestellt werden, dass nur die für die jeweilige Hardware passende Firmware verwendet wird.

4 Konzept für die Implementierung von *OTA*-Updates

Um die im vorherigen Abschnitt genannten Anforderungen an *OTA*-Updates zu erfüllen, wurde zunächst eine IoT-Infrastruktur realisiert, die den Build-Prozess für die Firmware, das zugehörige Repository sowie das WLAN-Netz mit den IoT-Endgeräten und deren Controller umfasst. Für die in diesem Paper verwendete Implementierung wurde bewusst auf

leichtgewichtige, gängige Software-Projekte gesetzt, um die Austauschbarkeit individueller Komponenten zu erleichtern. Abbildung 1 zeigt die verwendete IoT-Umgebung.

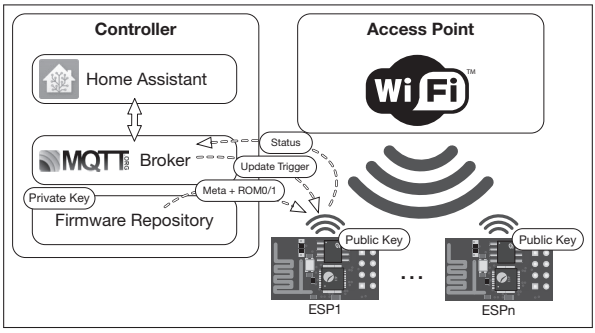


Abb. 1: Verwendete Netztopologie und Systemarchitektur.

Um die zentrale Verwaltung und Überwachung von Endgeräten zu ermöglichen, sendet jedes Endgerät Statusinformationen an ein vordefiniertes *MQTT* Topic, sobald es mit dem Netzwerk verbunden ist. Neben Informationen zum Typ des Endgeräts sowie Chip- und Flash-ID werden Details zum Bootloader, SDK und der derzeitigen Firmware-Version sowie relevante Angaben zur Bootloader-Konfiguration, wie z.B. der von der aktuell laufenden Firmware verwendete ROM-Bereich sowie der vom Boot-Prozess standardmäßig verwendete ROM-Bereich übermittelt. Dadurch können Administratoren Endgeräte mit veralteter Firmware finden, z.B. um fehlende oder fehlgeschlagene Updates zu erkennen.

4.1 Framework und Build-Infrastruktur

Das Framework umfasst ein Build-System, welches die Konfiguration von Basisparametern für alle Endgeräte erlaubt. Diese umfassen u.a. WLAN-Zugriffsparameter, *MQTT* Verbindungseinstellungen und URLs für den Bezug von Updates. Durch die Wiederverwendung von gleichen Code-Fragmenten wird sichergestellt, dass alle Endgeräte das gleiche Verhalten z.B. in Bezug auf die Übermittlung ihres Status oder die Interaktion mit dem Heimautomatisierungs-Controller aufweisen. Dies erleichtert die Konfiguration und erlaubt die Sammlung von Informationen zu den Endgeräten an einer zentralen Stelle.

4.2 Einrichtung des Endgeräts und Flash Layout

Auf der *ESP8266* MCU basierende Mikrocontroller-Boards haben meist das gleiche Layout: die MCU ist verbunden mit einem Flash-Speicher, der den Bootloader, die Firmware und andere Anwendungsdaten speichert. Das Memory Mapping der MCU ermöglicht nur jeweils eine 1 MB Memory Page aus dem Flash abzubilden [ES]. Zusätzlich muss der Zugriff auf die Blöcke auf die Blockgrenzen (je 1 MB) ausgerichtet werden. Da die herunterzuladende

Firmware größer als der freie Memory Heap Space sein kann, müssen die empfangenen Daten direkt in den Flash-Speicher geschrieben werden. Dabei kann die Ausführung von Code aus den gleichzeitig überschriebenen Memory Mapped Flash Pages zu unerwartetem Verhalten führen, da der Schreibvorgang den Code während der Ausführung verändert. Um dieses Problem zu verhindern wurde in dem Projekt der Flash-Speicher in zwei Hälften geteilt, wobei zwei ROM Slots entstehen, die unterschiedliche Firmware-Versionen beinhalten können (vgl. Abbildung 2). Eine davon wird z.B. aktuell ausgeführt, die andere kann z.B. durch den Download eines Firmware-Updates überschrieben werden. Neben den zwei Firmware ROM Slots bietet der Flash-Speicher Platz für den Bootloader und dessen Konfiguration. Für das Alignment und ein vereinfachtes Debugging wurde der zweite ROM Slot um die Größe des Bootloaders verschoben (Padding). Die 8192 Byte große Lücke kann für das persistente Speichern von Anwendungsdaten über Firmware-Updates hinweg genutzt werden. Der zusätzliche "standby" ROM Slot dient als Sicherungsmechanismus. Schlägt ein Update fehl oder wird unterbrochen, bleibt die vorherige Firmware-Version in dem "aktiven" Slot intakt und kann ohne Ausfall des Endgeräts weiter verwendet werden.

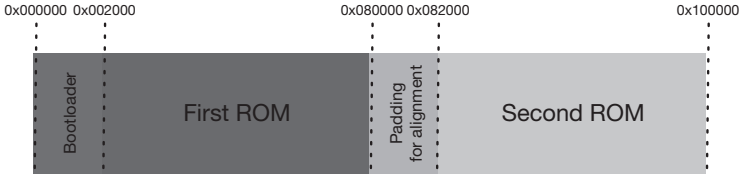


Abb. 2: Layout des Flash-Speichers zur Verwendung von zwei separaten ROM Slots.

4.3 Verschlüsselung der Firmware-Updates

Um Manipulationen und Fehler bei der Übertragung der Firmware auf das Endgerät zu verhindern, berechnet der Update-Prozess einen Hash der übermittelten Firmware und prüft diesen anhand der mitgesendeten digitalen Signatur. Hierfür wird der Hash-Algorithmus *SHA-256* [EH11] und ein Elliptic Curve Cipher basierend auf *Curve25519* [Be06] verwendet, wie u.a. derzeitig in [BD16; Fe17] als sicheres Verfahren für die Signatur von Software empfohlen. Die digitale Signatur wird während des Build-Prozesses mit Hilfe des privaten Schlüssels erzeugt und als Metainformation zum Update bereitgestellt. Als Gegenstück dazu nutzen die Mikrocontroller den zugehörigen öffentlichen Schlüssel für die Überprüfung der digitalen Signatur nach dem Empfang des Updates. Wie bereits im Abschnitt 4.2 erläutert, muss die neue Firmware direkt auf den Flash-Speicher geschrieben werden. Entsprechend wird der *SHA-256* Hash während des Downloads und dem Schreibvorgang auf den Flash-Speicher ermittelt. Nachdem der Download erfolgreich beendet wurde, wird der Hash-Wert anhand der digitalen Signatur verifiziert. Sofern die Überprüfung erfolgreich ist, wird die Bootloader-Konfiguration geändert und die neue Firmware aktiviert. Andernfalls bleibt die alte Firmware des Endgeräts aktiv und der Neustart entfällt.

5 Implementierung

Für *OTA*-Updates ist das Zusammenspiel verschiedener Komponenten bzw. Systeme nötig. Dabei übersetzt das Build-System den Quelltext und erzeugt die Firmware-Dateien inklusive zugehöriger digitaler Signaturen. Die Deployment-Infrastruktur stellt die Firmware-Dateien bereit und löst die Aktualisierung der Geräte aus. Die Implementierung des Update-Mechanismus als Teil der installierten Firmware des eingebetteten Systems ist selber verantwortlich für den Lade- und Installationsprozess der aktualisierten Firmware.

5.1 Build und Deployment

Sowohl das in diesem Paper vorgestellte ESPer Framework als auch das Build-System als solches unterstützen die Erzeugung von Firmware für unterschiedliche Geräteklassen. Dabei kontrolliert das Framework den Lebenszyklus der Firmware. Zu diesem Zweck definiert das Framework eine simple Schnittstelle, welche durch alle Geräteklassen implementiert werden muss. Dabei muss eine Instanz von *Device* erzeugt und zurückgegeben werden, welches selber wiederum Instanzen des Typs *Feature* enthält. Während die *Feature*-API es erlaubt Funktionalität durch Polymorphismus zur Laufzeit zu vereinheitlichen, wird bei der *Device*-Erzeugung Polymorphismus zur Übersetzungszeitpunkt eingesetzt um sowohl die Speicherverwaltung zu vereinfachen als auch virtuelle Funktionstabellen an dieser Stelle zu vermeiden. Abbildung 3 zeigt den kompletten gerätespezifischen Quellcode für eine simple schaltbare Steckdose, welcher sich auf die Definition des Typs und dessen Eigenschaften beschränkt (bspw. die zu nutzenden GPIO-Pins).

```
constexpr const char NAME[] = "socket";
constexpr const uint16_t GPIO = 12; // General purpose I/O

Device device;
OnOffFeature<NAME, 12, false, 1> socket(&device);

Device* getDevice() { return &device; }
```

Abb. 3: Gerätespezifischer Quellcode für eine schaltbare Steckdose.

Während der Übersetzung und Erzeugung der Firmware-Dateien werden auch deren Metainformationen als Datei in das jeweilige Verzeichnis gespeichert. Zusätzlich werden Konstanten aus der Build-Umgebung direkt in die resultierenden Firmware-Dateien geschrieben. Neben den WLAN-Zugangsdaten, den *MQTT*-Topics und weiteren konfigurierbaren Parametern stehen zusätzlich der Gerätetyp und die Firmware-Version als Konstanten zur Verfügung. Darüber hinaus wird auch der öffentliche Schlüssel zur Verifizierung der Firmware-Signaturen aus dem privaten Schlüssel generiert und als Objekt-Datei gegen jedes Firmware-Image gelinkt (Abbildung 4). Dies erlaubt die Nutzung aller Informationen im Quelltext ohne Einschränkungen obwohl diese erst zur Übersetzungszeit konfigurierbar bzw. bekannt sind. Der *ESP-01s* ist lediglich mit 1 MB Flash-Speicher ausgestattet, wobei dieser als einzelner kompletter Adressbereich zur Verfügung steht (siehe Abschnitt 4.2). Daher

kann der zweite ROM Slot nicht die gleiche Start-Adresse wie der erste ROM Slot nutzen. Da die Firmware ohne einen dynamischen Linking-Mechanismus ausgeführt wird und der ESP keinen positionsunabhängigen Code unterstützt, ist es nötig, dass die Adressierung je nach Offset der Firmware im ROM angepasst wird. Aus diesem Grund werden durch zwei Linker-Skripte die zwei Ausprägungen der Firmware erstellt, je nach Zielposition innerhalb des Speichers. Die zwei resultierenden Firmware-Images werden beide via *HTTP 1.1* zur Verfügung gestellt. Welches schließlich heruntergeladen und installiert wird hängt vom Zielslot ab. Abbildung 5 zeigt die Unterschiede der zwei Linker-Skripte, wobei *${SLOT}* die Slotnummer für den aktuellen Übersetzungsprozess enthält.

```
update_key_pub.bin:
    echo "$(UPDATE_KEY)" | ecdsakeygen -p | xxd -r -p > "$@"

update_key_pub.o: update_key_pub.bin
    $(OBJCOPY) -I binary $< -B xtensa -O elf32-xtensa-le $@
```

Abb. 4: Erzeugung der Objekt-Datei für den öffentlichen Schlüssel.

```
irom0_0_seg :
  org = ( 0x40200000          // The memory mapping address
        + 0x2010              // Bootloader code and config
        + 1M / 2 * ${SLOT} ), // Offset for the ROM slot
  len = ( 1M / 2 - 0x2010  )  // Half ROM size excl. offset
```

Abb. 5: Linker Skript zur Erstellung der Firmware für die zwei unterschiedlichen ROM Slots.

Der Erstellungsprozess erzeugt neben den zwei Firmware-Images auch die dazugehörigen Dateien mit den Metainformationen. Für deren Erzeugung wird die aktuelle Version in eine Datei *.version* geschrieben. Nach der Fertigstellung werden die Signaturen der beiden Firmware-Dateien erzeugt und den Dateien entsprechend angehängt. Nach der erfolgreichen Erzeugung der Firmware und der zugehörigen Dateien (Metainformationen) für alle Geräteklassen, werden diese auf den Repository-Server kopiert, wo diese mittels *HTTP 1.1* zur Verfügung gestellt werden. Dieser wird auch in der Konfigurationsdatei des Projekts angegeben und gilt für die Geräte als Quelle für Aktualisierungen.

5.2 Update-Prozess

Der Aktualisierungsvorgang besteht aus vier Phasen: Prüfung auf Aktualisierungen, Reprogrammierung des Geräts, Berechnung und Verifizierung der digitalen Signaturen der zu installierenden Firmware und schließlich - im Falle der erfolgreichen Aktualisierung - der Neukonfiguration des Boot-Prozesses zur Nutzung der neue Firmware. Um die IoT-Geräte über die Verfügbarkeit neuer Firmware-Versionen zu informieren hält der Update-Server für jede Geräteklasse eine Datei mit den Metainformationen zur letzten verfügbaren Firmware-Version. Diese Metainformationen enthalten sowohl die Versionsnummer als auch die digitalen Signaturen beider Firmware-Dateien. Diese Informationen werden durch den Update-Server mit Hilfe von *HTTP 1.1* als *${DEVICE}.version* zur Verfügung gestellt

(wobei *${DEVICE}* jeweils durch die Geräteklasse substituiert wird). Jedes Gerät fragt regelmäßig den durch die *UPDATER_URL* spezifizierten Update-Server nach der verfügbaren Firmware-Version. Dabei werden die Metainformationen heruntergeladen und die Versionsnummer mit der aktuell installierten Version verglichen. Falls sich diese unterscheiden wird der eigentliche Aktualisierungsprozess initiiert. Sofern Fehler beim Download auftreten, wird der Versuch beim nächsten Überprüfungsintervall wiederholt. Abbildung 6 zeigt die Bestimmung der Download-Adresse und die Neukonfiguration des Bootloaders. Dabei stellt der Update-Server die Firmware-Dateien analog zu den Metainformationen zur Verfügung. Durch Ergänzung des Pfades mit *.rom{0,1}* kann somit auf das Firmware-Image für den ersten bzw. zweiten ROM-Slot zugegriffen werden. Die gewählte Datei wird dann entsprechend über *HTTP 1.1 GET* vom Update-Server heruntergeladen.

```
#define URL_ROM(slot) (( URL "/" DEVICE ".rom" slot ))

// Select rom slot to flash
const auto& bootconf = rboot_get_config();
if (bootconf.current_rom == 0) {
  updater.addItem(bootconf.roms[1], URL_ROM("1"));
  updater.switchToRom(1);
} else {
  updater.addItem(bootconf.roms[0], URL_ROM("0"));
  updater.switchToRom(0); }
```

Abb. 6: Download-Adresse und Neukonfiguration des Bootloaders je nach verwendetem ROM-Slot.

Die Firmware wird in Teilstücken vom Update-Server heruntergeladen. Dabei wird zunächst die *SHA256* Checksumme aktualisiert bevor das Teilstück in den Flash-Speicher geschrieben wird. Nach dem erfolgreichen Schreiben wird mit dem nächsten Teilstück fortgefahren. Bei erfolgreichem Abschluss des Vorgangs wird der resultierende Hashwert gegen die Signatur des Firmware-Images geprüft. Dazu wird der kryptografisch signierte Hashwert aus den Metainformationen gegen den *Curve25519* öffentlichen Schlüssel der installierten Firmware geprüft. Nur wenn die Checksummen mit der gegebenen Signatur übereinstimmen wird die Firmware als valide angenommen und der Aktualisierungsprozess fortgesetzt. Als Bootloader wird *rBoot* [Bu] verwendet, da dieser im *Sming* Framework integriert ist und unterschiedliche ROM-Slots booten kann. Zur Konfiguration muss eine *rBoot*-spezifische Struktur an eine definierte Stelle im Flash geschrieben werden. Diese Struktur enthält die Ziel-Offsets für alle bekannten ROM-Slots und die Nummer des ROM-Slots der während des Bootvorgangs verwendet werden soll. Um diesen nach einer erfolgreichen Aktualisierung zu wechseln wird die Struktur angepasst und ein Neustart des Geräts initiiert.

Falls die die kryptografische Signatur nicht validiert werden kann wird die aktuelle Konfiguration beibehalten und der Neustart entfällt. Ein weiterer Update-Versuch wird dann nach Ablauf des eingestellten Intervalls oder durch erneute Signalisierung eines verfügbaren Updates unternommen.

6 Fazit und Ausblick

In diesem Paper wurde ein Konzept für die Erzeugung und Verteilung von kryptographisch gesicherten *Over-the-Air*-Updates für IoT-Endgeräte basierend auf dem ESP8266 Mikrocontroller vorgestellt. Die entwickelte Proof of Concept Implementierung ist essentieller Bestandteil des Home-Automation Development und Deployment im *Magrathea Laboratories e.V.* Hackerspace [Ma]. Alle Endgeräte, die in dieser Umgebung eine *OTA*-unterstützende Firmware verwenden wurden mit der hier vorgestellten Lösung mehrfach erfolgreich und ohne Probleme im laufenden Betrieb aktualisiert, ohne dass eine manuell Intervention erforderlich war. Während des Updates war die Funktion der Geräte nicht eingeschränkt und sie konnten bis zum Zeitpunkt des Updates wie gewohnt verwendet werden. Hierbei wurden auch Änderungen an der Netzkonfiguration und wichtige Stabilitätsupdates am Netzwerk-Stack durchgeführt. Durch die Entwicklung des in diesem Paper vorgestellten Frameworks für einen automatisierten Build- und Update-Prozess wird dieses Problem adressiert und eine einheitliche Umgebung für die sichere Verteilung von Firmware-Updates für IoT-Endgeräte bereitgestellt. Dadurch konnte der Entwicklungsprozess vereinfacht werden, Änderungen im Code schneller in abhängige Module und Funktionen übernommen werden und eine zeitnahe Verteilung auf die Endgeräte im Sinne einer Continuous Delivery erzielt werden. Zukünftig soll die Funktionalität und Sicherheit des Frameworks zusätzlich erweitert werden. Die aktuelle Entwicklung umfasst weitergehende Sicherheitsmechanismen wie die Verifikation der Integrität der Firmware während des Boot-Vorgangs, um Manipulationen oder defekte Firmware-Images zu erkennen. Darüber hinaus wird die Aufnahme des Endgerätetyps in die digitale Signatur in Betracht gezogen, um Fehlzuordnungen der Firmware zu nicht unterstützten Endgeräten zu verhindern. Ebenfalls ist eine Übernahme der Firmware in den "standby" ROM-Slot nach jedem erfolgreichen Update geplant, um die Fehlertoleranz weiter zu steigern. Zusätzlich ist die Erweiterung der Statusinformationen und die Entwicklung einer Web-Anwendung für das Monitoring der IoT-Infrastruktur in der Umsetzung.

Literatur

[BD16] Barker, E.; Dang, Q.: NIST Special Publication 800–57 Part 1, Rev. 4, 2016.

[Be06] Bernstein, D. J.: Curve25519: new Diffie-Hellman speed records. In: International Workshop on Public Key Cryptography. Springer, S. 207–228, 2006.

[Be16] Beresford, A. R.: Whack-A-Mole Security: Incentivising the Production, Delivery and Installation of Security Updates. In: IMPS@ESSoS. S. 9–10, 2016.

[Bu] Burton, R. A.: An open source bootloader for the ESP8266, `https://github.com/raburton/rboot`, [abgerufen am: 2018-04-16].

[Di15] Di Nisio, A.; Di Noia, T.; Carducci, C. G. C.; Spadavecchia, M.: Design of a low cost multipurpose wireless sensor network. In: 2015 IEEE International Workshop on Measurements and Networking (M&N). IEEE, S. 1–6, 2015.

[EH11] Eastlake, D.; Hansen, T.: US Secure Hash Algorithms (SHA and SHA-based HMAC and HKDF), RFC 6234, http://www.rfc-editor.org/rfc/rfc6234.txt, [abgerufen am: 2018-04-16], RFC Editor, Mai 2011.

[ES] ESP8266 Community: ESP8266 Memory Map, http://www.esp8266.com/wiki/doku.php?id=esp8266_memory_map, [abgerufen am: 2018-04-16].

[Fe17] Federal Office for Information Security: Cryptographic Mechanisms: Recommendations and Key Lengths, BSI – Technical Guideline BSI TR-02102-1, https://www.bsi.bund.de/SharedDocs/Downloads/EN/BSI/Publications/TechGuidelines/TG02102/BSI-TR-02102-1.pdf, [abgerufen am: 2018-04-16], Federal Office for Information Security, Feb. 2017.

[Go16] Gore, S.; Kadam, S.; Mallayanmath, S.; Jadhav, S.: Review on Programming ESP8266 with Over the Air Programming Capability. International Journal of Engineering Science 3951/, 2016.

[KS16] Kodali, R. K.; Soratkal, S.: MQTT based home automation system using ESP8266. In: 2016 IEEE Region 10 Humanitarian Technology Conference (R10-HTC). IEEE, S. 1–5, 2016.

[Ma] Magrathea Laboratories e.V.: Magrathea Laboratories - Creating new Worlds, https://maglab.space/, [abgerufen am: 2018-04-16].

[MFE12] Mansour, K.; Farag, W.; ElHelw, M.: AiroDiag: A sophisticated tool that diagnoses and updates vehicles software over air. In: 2012 IEEE International Electric Vehicle Conference (IEVC). IEEE, S. 1–7, 2012.

[NL08] Nilsson, D. K.; Larson, U. E.: Secure Firmware Updates over the Air in Intelligent Vehicles. In: ICC 2008 - 2008 IEEE International Conference on Communications Workshops. IEEE, S. 380–384, 2008.

[NS16] Nigappa, K.; Selvakumar, J.: Industry 4.0: A Cost and Energy efficient Micro PLC for Smart Manufacturing. Indian Journal of Science and Tech. 9/44, 2016.

[Ru16] Ruckebusch, P.; De Poorter, E.; Fortuna, C.; Moerman, I.: GITAR - Generic extension for Internet-of-Things Architectures enabling dynamic updates of network and application modules. Ad Hoc Networks/, 2016.

[Th16] Thaker, T.: ESP8266 based implementation of wireless sensor network with Linux based web-server. In: 2016 Symposium on Colossal Data Analysis and Networking (CDAN). IEEE, S. 1–5, 2016.

[We16] Weisbach, M.; Taing, N.; Wutzler, M.; Springer, T.; Schill, A.; Clarke, S.: Decentralized coordination of dynamic software updates in the Internet of Things. In: 2016 IEEE 3rd World Forum on Internet of Things (WF-IoT). IEEE, S. 171–176, 2016.

[WKM16] Walia, N. K.; Kalra, P.; Mehrotra, D.: An IOT by information retrieval approach: Smart lights controlled using WiFi. In: 6th International Conference - Cloud System and Big Data Engineering (Confluence). IEEE, S. 708–712, 2016.

IT-Management, Lehre und Testbeds

Müller, Neumair, Reiser, Dreo Rodosek (Hrsg.): 11. DFN-Forum Kommunikationstechnologien,
Lecture Notes in Informatics (LNI), Gesellschaft für Informatik, Bonn 2018 81

Establishing a Universal Service Model for Authentication Scenarios based on MNM Service Model

Jule Anna Ziegler[1]and David Schmitz[1]

Abstract: In today's diversity and heterogeneity of authentication protocols, standards, technologies and frameworks it is hard to compare or at all to combine, e.g., for multi factor authentication, different authentication scenarios. Above all, they all have different understanding or at least perspective on the term "service". This perspective on a service is normally only a technically one, not really taking into account a full IT service covering the whole life cycle from design, (trust) negotiation, operation to decommissioning. This paper proposes a Universal Authentication Service Model (UASM) in order to describe authentication scenarios in a generic scenario-independent service-oriented way. Requirements are derived from seven example authentication scenarios. It outlines the characteristics of the MNM Service Model (MSM) and how this approach can be reused for universally describing any authentication scenario. Since the work described here is still work in progress, in this paper, UASMs main terms and concepts are introduced. Finally, by using step-by-step refinement of MSM Basic View, UASMs Basic Views are described.

Keywords: Authentication, Identity Management, Service Management, MNM Service Model

1 Introduction

Today's landscapes consist of IT services of various types, webservices, e.g., social networks, cloud storage, but not limited to, also intra/inter-organizational services like email or collaborating tools, all requiring authentication and subsequent authorization to access the respective service. Authentication, in short **AuthN**, constitutes to be the verification of the authenticity of the entity or subject to be authenticated, e.g., user or thing in IoT, the permission about accessing a service is authorization, **AuthZ**. In the area of AuthN, which is the focus of this paper, many different AuthN protocols, standards, technologies and frameworks (abbreviated with **AuthN P/S/T/Fs**) have emerged.

Hence, before establishing a new or enhancing an existing AuthN service scenario, e.g. especially in terms of multi factor authentication, we think organizations profit by identifying at first existing AuthN components, roles, processes and their dependencies in place. Above all, a *complete* and *comparable* description of planned or existing AuthN service scenarios is also relevant with regard to performing analysis, in particular security analysis, but also in consideration of other criteria, such as, for example, effort or costs. Comparability is achieved by a uniform terminology whereas completeness is achieved by a holistic description of all processes and roles in a service (lifecycle) comprehensive way. Furthermore, many AuthN P/S/T/Fs have a different understanding or at least perspective on the term "service". This perspective on a service is normally only a technical one, not

[1] Leibniz Supercomputing Centre, Boltzmannstr. 1, 85748 Garching n. Munich, {surname}@lrz.de

really taking into account a full IT service covering the whole lifecycle from design, (trust) negotiation, operation to decommissioning.

Especially in regard to multi factor authentication, AuthN services are more and more composed out of different dependent AuthN services, e.g. SAML password AuthN realized by LDAP combined with Universal Second Factor (U2F) protocol for second factor AuthN. When dealing with concrete AuthN service scenarios, dependent services which use an AuthN service can be of interest as well. That is why, in this work an **AuthN (service) scenario** is a scenario, which includes at least one AuthN service which may be used by other dependent services and may itself be dependent or internally composed out of simpler AuthN services. Thereby, in order to perform holistic analysis, e.g., regarding security of a particular AuthN factor, for all involved AuthN services all service processes (e.g., proper authentication, registration, incident management) and their dependencies need to be identified and specified.

A proper basis for the overall description of service interactions are UML Use Case Diagrams and UML Activity Diagrams. Thus, there is a uniform language for interactions/processes, but service dependencies or, for example, relationships between roles and services are still missing.

Since many organizations and campuses in higher research & education participate in the service *eduGAIN* provided by GÉANT, they are most often faced with multiple AuthN P/S/T/Fs which can become challenging. E.g. taking into account various AuthN P/S/T/Fs like Security Assertion Markup Language (SAML) or OpenID Connect, entities acting in multiple roles but also considering involved resources like for instance an LDAP database. Above all, many AuthN P/S/T/F are using their own, diverse vocabulary, terminology and concepts (e.g. SAML Service Provider vs. OIDC Relying Party, SAML attributes vs. OIDC claims).

Summarized, we identified the following problem statements:

- How to describe AuthN scenarios in a generic scenario-independent service-oriented way?
- What are the requirements for such a model?
- Is it possible to reuse/extend an existing approach?

To the best of our knowledge, models for describing AuthN scenarios in a generic but also scenario-independent way and simultaneously covering the full IT service lifecycle do not exist and do not address to the above-mentioned problem statements.

The remainder of this paper is structured as following: In section 2 we derive requirements from seven example scenarios. Section 3 outlines characteristics of MNM Service Model and why this approach can be reused for universally describing AuthN service scenarios. Following, in section 4 the refinement of MNM Service Model for AuthN service scenarios by introducing main terms and concepts is described. The last section 5 concludes the paper and since the work described in this paper is still work in progress an outlook to future work is provided. A list of abbreviations is attached at the very end of this paper.

2 Authentication Service Scenarios and Requirements

For requirement gathering, we use both universally valid AuthN service scenarios but also specific AuthN service scenarios occuring within eduGAIN representing different AuthN P/S/T/Fs. Here, AuthN service scenarios are exemplary introduced from the point of view of the service lifecycle phase *operation*, which implies that service establishment (especially design, trust negotiation) has already taken place.

2.1 Authentication Service Scenarios

eduGAIN [GÉ18] provided by GÉANT is an identity inter-federation which interconnects (national) identity federations to facilitate access to resources and services (e.g., wiki, survey tools, e-Learning) needed by the global research and education community. Based on a federated identity, which is typically assigned by the user's home organization, e.g., university, school, research center, users can access various external services provided by eduGAINs participating service providers. While AuthZ is performed by the service provider itself, AuthN is performed by the respective home organization of the user. Here, AuthN is mostly not limited to one specific AuthN P/S/T/F.

In this section we describe seven possible AuthN scenarios, SC0 to SC6, including single factor AuthN scenarios, i.e., using only one factor of something you know/have/are but also multi factor AuthN (in short: MFA) scenarios using at least two of the four different factor types. In particular, such combination arises for the case of the example scenarios SC4 to SC6. Before, scenarios SC0 to SC3 represent examples for applying a single particular AuthN P/S/T/F.

SC0 (Scenario 0) represents a simple AuthN scenario, where a user authenticates to a local/not distributed system, e.g., an employee enters his/her password to log on to the system of his/her working station. Here, AuthN is performed by the system itself, by comparing username and password with the corresponding attributes stored in the organizational internal LDAP database.

SC1 describes AuthN within a SAML (inter-)federation. While trying to access a particular web service, the user, in SAML called principal, is redirected by the SAML Service Provider (SAML SP) via discovery service to its home SAML Identity Provider (SAML IdP). After successful AuthN at the SAML home IdP, for example by means of username and password, the SAML home IdP responds with an AuthN response which serves as basis for the SAML SPs authorization decision.

SC2 outlines AuthN using OpenID Connect (OIDC). If a user wants to log on to a service, the OIDC Relying Party (OIDC RP) generates an AuthZ request which is sent, using a discovery service (e.g., Webfinger) to the corresponding OpenID Provider (OIDC OP) of the user, also including the additional parameter scope=openid to indicate that user AuthN at the OIDC OP is required. The identity of the user and information about the AuthN is transferred in form of an ID token from OIDC OP to OIDC RP.

In **SC3**, U2F protocol is used for second factor AuthN with a USB, NFC oder Bluetooth device in connection with a U2F server.

While the four previous examples scenarios presented so far apply only to a single AuthN P/S/T/F, e.g., either SAML, OIDC, or U2F, the following ones, all concerned with MFA, will at least be a combination of two AuthN P/S/T/Fs and so demonstrating the need for universal, generic terms and concepts.

SC4 is a MFA scenario within a SAML-based federation using a proxy pattern. The Dutch Identity Federation SURFnet is implementing a Hub&Spoke federation architecture, where all SAML home IdPs and SAML SPs are connected via a central hub, a SAML proxy. Between the central hub and the SAML SPs sits their so called Step-up Authentication as a Service [va17], also a (transparent) SAML proxy, which offers 2nd factor IdPs for each supported 2nd factor type, e.g., YubiKey, Tiqr. In case a SP sends an AuthN request, the Step-up Authentication Service requests 1st factor AuthN from the home IdP, then requests second factor AuthN from the respective Step-up 2nd factor IdP, then the proxy sends an AuthN response to the SAML SP.

SC5 represents an integrated MFA scenario in a SAML federation, where first factor AuthN is provided by the SAML IdP, second factor AuthN is performed within the SAML IdP internally, e.g., U2F, which requires high adaption in IdP realization/implementation.

SC6 is based on a SAML federation for first factor AuthN, but second factor AuthN integration is done by the SAML SP instead of the SAML IdP. Hence, second factor AuthN is decoupled from first factor AuthN, which needs high adaption in each SPs realization/implementation.

As we can see from the diversity and heterogeneity of today's AuthN service scenarios, especially in regard to terminology and concepts, there is a lack of a unique generic model, which allows a comparable, complete and common understanding of AuthN service scenarios.

2.2 Requirements

Based on the scenarios described above and the problem space outlined in section 1, we derived the following requirements for describing AuthN scenarios:

- **Generic and universal AuthN terminology:** The model should provide a universal, uniformly applicable AuthN terminology, leading to better understanding between all involved roles, i.e. customers, users, and providers of any involved services, but also regarding different AuthN P/S/T/Fs being used for AuthN. Maybe also in parallel in a single given scenario.
- **Consideration of usage functionality** (user aspects) and **management functionality** (customer aspects, including trust negotiation/mangement, registration of each factor and user centric right management).
- **Recursive application** of the model in order to support nested hierarchies of AuthN, e.g. for MFA.

- **Scenario-independent modeling templates:** Any abstract or specific AuthN scenario is representable by using scenario-independent modeling templates.

3 Munich Network Management (MNM) Service Model

We checked existing models, such as tmforum Information Framework (SID) [tmf18] and MNM Service Model [Ga01a, Ga01b, Ga02] against the requirements in section 2 and identified the MNM Service Model as a suitable approach to describe AuthN scenarios in a generic scenario-independent service-oriented way. In this section we outline the characteristics of the MNM Service Model, abbreviated in the following with **MSM**.

MSM proposes a systematic methodology used for analyzing and identifying actors of services including their inter/intra-organizational relationships. For this, MSM introduces a generic scenario-independent service model which covers the whole lifecycle of IT services. MSM distiguishes between three different views: **MSM Basic Service Model** (here, unified into **MSM Basic View**) identifies roles, i.e. user, customer and provider, and their associations. Based on this as a refinement **MSM Service View** describes services independently from its service implementation always differentiating between customer side, provider side and side independent aspects of a service. **MSM Realization View** represents the provider-internal realization of IT services. MSM uses an abstraction of interaction classes by classifying interactions in usage and management (functionality) and also supports chaining of MSM, a recursive application of services, i.e., sub services.

We decided to use MSM due to its suitable and useful language with generically, commonly defined terms / model specification formalism between customer and provider side (MSM Service View). But also because of its specification of internal realization at provider side (MSM Realization View) and its splitted consideration into usage functionality (user aspects) and management functionality (customer aspects, including trust negotiation and management). Moreover, MSM Basic View allows the consideration of entities, roles and services along with their sub service relationships.

4 Using MNM Service Model for AuthN Service Scenarios

In this section, we refine the MNM Service Model (MSM) in order to achieve a generic scenario-independent service-oriented model for AuthN scenarios. The existing MSM with all its generic classes and associations/relationships in Basic/Service/Realization View is referenced hereafter as generic MSM, while its refinement to AuthN scenarios, i.e., refined classes, associations/relationships specifically for AuthN, is named **Universal Authentication Service Model (UASM)**. For UASM, we propose to describe, specify, and model **AuthN service scenarios**, i.e., IT service scenarios involving AuthN P/S/T/Fs like SAML, OIDC, Radius or U2F (see scenarios SC0 to SC6) but also in general, independently of used AuthN P/S/T/F, based on MSM. We restrict ourselves in this paper to the step-by-step refinement of MSM Basic View towards four different types of UASM Basic Views that can be used for any AuthN scenario. Before, main terms and concepts are introduced.

While MSM also could be applied for each particular AuthN P/S/T/F-specific service scenario singly and independently, e.g., particular for a SAML or OIDC-based AuthN service, here we first abstract from the concrete AuthN P/S/T/F to yield a refinement of MSM. That is, refined classes and maybe based on that, refined associations/relationships in MSM Basic/Service/Realization View, which can in turn be instantiated to any AuthN service scenario.

The refinement for UASM is based on the generic concept of **AuthN Information (AuthNI)**. Here, AuthNI is defined as an information about the authentication of a real-world identity, in UASM called **Real-World AuthN Subject (RAS)**. In most of the use cases the RAS is typically a human, in other cases it can be also whole organizations or divisions of it (technical accounts). Considering Internet of Things a real-world ”Thing” is possible, too. AuthNI includes in particular the **Trustworthy AuthN Subject (TAS)**, which is the digital counterpart of RAS, both being subsumed under the term (AuthN) subject. So, AuthNI is actually an information about the AuthN of the TAS. In addition to the TAS, AuthNI may include a multitude of fixed and dynamic AuthN attributes, also including meta attributes, e.g., name of user, email address of user, number and type of factors used for AuthN, timing, but also the time of the last change of a factor, strength of the factors actually used.

To demonstrate the usage of the term ”AuthNI” let’s regard two examples: On the one hand, AuthNI can appear and be used internally by an entity, e.g., in SC1, SC4-SC6 at a SAML IdP when comparing user-provided AuthNI (e.g., first factor credentials) to registered values in a local resource used for verification of user-provided AuthNI, e.g., an LDAP directory. Here, the LDAP directory is a resource for realizing the trusted AuthNI provisioning in the SAML IdP. On the other hand, AuthNI is often communicated between entities in a trusted way, e.g., in SC1-SC2, SC4-SC6, between a SAML IdP and SAML SP, or between an OIDC OP and OIDC RP. In this case, the receiving entity trusts the sending one in order to get a trusted and trustful statement about the AuthN of the respective Real-World AuthN Subject (RAS) via its digital counterpart (TAS). Both cases are covered by the term AuthNI, not differentiating what the particular (possibly meta) attributes are. It may contain the plain text password (ideally only if used internally to an entity), only the hash of the password, or only a statement that the password has been (successfully) checked (together with some verifiable context for the performed AuthN in that case), and/or similiar information about other factors beyond first-factor passwords.

If communicated between trusted entities AuthNI normally is enriched by means to ensure the trustfulness (i.e., to ensure authenticity, integrity) of it, typically by employing cryptographic means for the envelope of the AuthNI used (e.g., certificates, signatures, hashes) or for the communication channel used to transmit AuthNI (e.g., encryption, certificates). This requires that such communicating entities establish and manage appropriate trust between themselves in advance. In this context the AuthNI can be more specifically named **Trustworthy AuthNI (TAuthNI)**.

UASM introduces as a refinement to MSM two kind of sub classes of (generic) services: **Trustworthy AuthNI Administration Service (TAAS)** and **Trustworthy AuthNI Provisioning Service (TAPS)**, both dealing with TAuthNI.

The former service type, TAAS, is the original or principal source of AuthNI, so it is responsible for the registration and administration of AuthNI, especially mapping digital identities to real-word identities. This includes the administration of fixed and dynamic AuthN attributes, also including (typically dynamic) meta attributes, like the time of the last change of a factor or the last time of the verification of a factor.

The latter service type, TAPS, is used for communicating TAuthNI, which is known to him - either from local resources realizing AuthN functionality and so sharing these with an corresponding TAAS or recursively from another TAPS, i.e., as a sending entity to another receiving entity. The main focus in this paper is on the TAPSs independent of the used AuthN P/S/T/F, while the TAASs are introduced to provide a complemented view on AuthNI regarding its principal origin.

This functionality of TAPS and TAAS is more concretized by splitting the functionality of each service into **usage** and **management functionality**: Interactions needed by the user to fulfill the purpose of the respective service, e.g. at the TAPS the authentication itself, are called usage functionality. Interactions for managing the service, such as the trust negotiation or the provisioning of an interface to report incidents, is subsumed under management functionality.

For shortness, an entity acting as a provider of TAPS is named **Trustworthy AuthNI Provider (TAP)**, and an entity acting as a customer of TAPS is named **Trustworthy AuthNI Consumer (TAC)**. Always depending on the specific AuthN scenario, here, for sake of simplicity, the TAP acts as both the provider of the TAPS and the TAAS.

4.1 $\text{UASM}_{\text{generic}}$ Basic View

After analyzing the AuthN scenarios of section 2 (predominantly eduGAIN AuthN scenarios), we discovered two main characteristics:

- There are high-level services which rely on the TAuthNI communicated by the TAPS. So, the user of the relying service is also user of the TAPS.
- Explicitly the RAS is user of the TAPS.

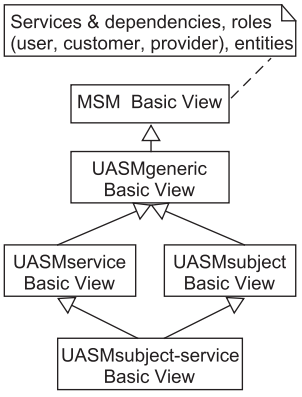


Figure 1: Step-by-Step refinement

Consequently, the RAS is also user of the relying (high-level) service. However, there are AuthN scenarios aside from eduGAIN being less specific, which should be covered by UASM as well, e.g. AuthN of Things in IoT. Therefore, the general idea was to build the model step-by-step (see Figure 1). $\text{UASM}_{\text{subject-service}}$ Basic View in Figure 5 covers exactly eduGAIN AuthN scenarios. By looking at the two characteristics individually, we get two more generic cases also covered by UASM. They are described in section 4.2 ($\text{UASM}_{\text{service}}$ Basic View) and section 4.3 ($\text{UASM}_{\text{subject}}$ Basic View).

With the help of the introduced terms and concepts, we introduce at first **UASM$_{\text{generic}}$ Basic View** (see Figure 2), which is based on MSM Basic View. UASM$_{\text{generic}}$ Basic View is the very fundamental structure and will be used for further refinement to meet the characteristics described above. Services are depicted as ellipses whereas (legal) entities as rectangles, roles (user, customer, provider) like UML association roles and dependencies between services as dashed arrows. Comments like in UML.

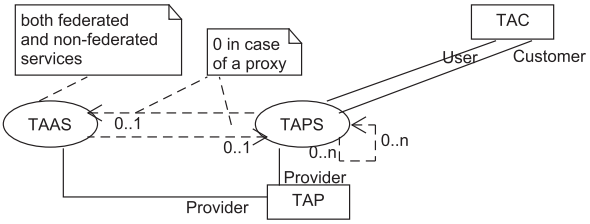


Figure 2: UASM$_{\text{generic}}$ Basic View

In UASM$_{\text{generic}}$, the TAPS serves as a completely generic TAPS and represents an abstract top level class without considering dependent high-level services. Also, in UASM$_{\text{generic}}$ a potentially human real-world subject (RAS) and its digital counterpart TAS, which are to be authenticated, are not considered as acting entities (in the sense of a user/customer) as far as the model and its AuthN services are concerned. That is why they are not present in Figure 2.

TAPSs can be chained, potentially in nested hierarchies with **high-level** TAPSs, e.g., SAML-Proxies, depending on **low-level** ones. In the case of proxies, the providing entity does typically not provide a corresponding TAAS.

The TAC is the user of the TAPS and depending on the AuthN scenario either a **direct** or an **indirect** customer of the TAPS. For example in eduGAIN, a SAML SP of one federation typically does not share a common bilateral contract with a SAML IdP of another federation and thus embodies an indirect customer relationship.

4.2 UASM$_{\text{service}}$ Basic View

Often in an AuthN scenario, e.g., in the case of SAML or OIDC, TAPS are **used as sub services** for a dependent high-level service. To differentiate this case from UASM$_{\text{generic}}$, it is called **UASM$_{\text{service}}$**.

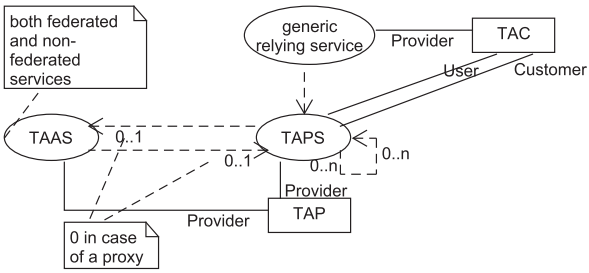


Figure 3: UASM$_{\text{service}}$ Basic View

In UASM$_{\text{service}}$, the dependent high-level service is called **TAuthNI-relying service** or **generic relying service** and is provided by the TAC (see Figure 3). The generic relying service in turn depends on the TAPS. Here, anything can be authenticated, without being explicit user (/customer) of the generic relying service or TAPS respectively. E.g., animals within an animal shelter, given that the animal does not use/manage authentication itself.

4.3 $UASM_{subject}$ Basic View

In the case called **$UASM_{subject}$**, illustrated in Figure 4, the RAS is explicitly seen as user of the TAAS and TAPS without considering dependent high-level services.

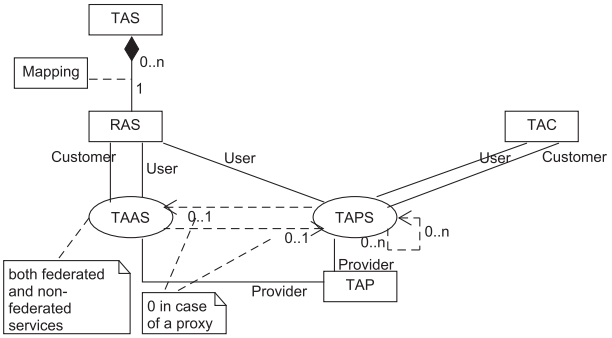


Figure 4: $UASM_{subject}$ Basic View

4.4 $UASM_{subject\text{-}service}$ Basic View

Often, in such AuthN scenarios the authenticated real-world identity is a human user which is the actual user (/customer) of the high-level service. This is especially the case in SAML, as in SC1, SC4-SC6 and was one of the main motivation to design for and test with UASM. For explicitly such AuthN service scenarios, $UASM_{generic}$ is first refined to $UASM_{service}$ / $UASM_{subject}$ and than extended to **$UASM_{subject\text{-}service}$** (see Figure 5). As a further refinement in this case, the dependent high-level service is called here **TAuthNI-relying subject-service (TRSS)**.

Here, the real-world identity is user (/customer) of the TRSS which is depending on the low-level TAPS. That is why, the real-world identity is recursively user/customer (either transparent or intransparent) of the low-level TAPS.

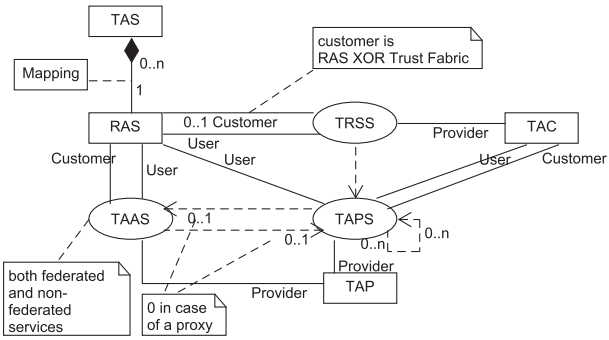


Figure 5: $UASM_{subject\text{-}service}$ Basic View

Moreover, in $UASM_{subject\text{-}service}$ scenarios the real-world identity is typically also explicit user and customer of a TAAS corresponding to the original/principal source of AuthNI regarding his digital identity. So the generic introduction of TAAS as a service (abstraction) shows its usefulness especially in this case.

As being based on MSM, all these service-entity-relationships can be specified and modeled explicitly in $UASM_{subject\text{-}service}$, either in MSM Basic View or recursively applied in MSM Service View/Realization View. Thereby, after having introduced the overall UASMs generic modeling scheme (general UASM definitions and refined MSM classes), Figure 5 provides the resulting $UASM_{subject\text{-}service}$ Basic View in MSM Basic View notation.

5 Conclusion and Further Work

In this paper we highlighted the importance of a generic scenario-independent service-oriented model for AuthN scenarios. For this purpose, seven AuthN scenarios were used to derive requirements, which in turn were used to examine to what extent an existing approach could be reused. The characteristics of the MNM Service Model (MSM) were outlined and shown to match very well to the requirements derived from the example AuthN scenarios. Following, main terms and concepts of the Universal Authentication Service Model (UASM), based on MSM, were introduced. The step-by-step refinement of MSM Basic View towards $\text{UASM}_{\text{generic}}$, $\text{UASM}_{\text{service}}$, $\text{UASM}_{\text{subject}}$ and $\text{UASM}_{\text{subject-service}}$ Basic View were depicted, which can be used to model and specify any AuthN scenario at an overview level.

Since the work described in this paper is still work in progress, but also due to space constraints, future work will show the instantiation of the example AuthN scenarios using UASM Basic View, which is based on MSM Basic View. Then, as a refinement of UASM Basic View, UASM Service View and UASM Realization View will be introduced. In addition to that, UASMs main terminology and concepts will be complemented, e.g., by the concept of external versus internal TAPS and the way to describe chaining of TAPS. Furthermore, more AuthN scenarios as well as on a higher refinement level need to be instantiated to show UASMs applicability, especially in regard to UASM Service View and UASM Realization View. Additionally, predefined technology specific UASM templates as building blocks for easier application need to be addressed.

References

[GÉ18] GÉANT. eduGAIN Homepage, 2018. `https://www.geant.org/Services/Trust_identity_and_security/eduGAIN`, accessed: 10/01/2018.

[Ga01a] Garschhammer, M.; Hauck, R.; Hegering, H.-G.; Kempter, B.; Radisic, I.; Rolle, H.; Schmidt, H.; Hegering, H.-G.; Langer, M.; Nerb, M.: Towards generic service management concepts a service model based approach. In: 2001 IEEE/IFIP International Symposium on Integrated Network Management Proceedings. Integrated Network Management VII. Integrated Management Strategies for the New Millennium (Cat. No.01EX470). IEEE, pp. 719–732, 2001.

[Ga01b] Garschhammer, M.; Hauck, R.; Kempter, B.; Radisic, I.; Roelle, H.; Schmidt, H.: The MNM service model - Refined Views on Generic Service Management. Journal of Communications and Networks, 3(4):297–306, dec 2001.

[Ga02] Garschhammer, M.; Hauck, R.; Hegering, H.-G.; Kempter, B.; Radisic, L.; Roelle, H.; Schmidt, H.: A case-driven methodology for applying the MNM service model. In: NOMS 2002. IEEE/IFIP Network Operations and Management Symposium. ' Management Solutions for the New Communications World'(Cat. No.02CH37327). IEEE, pp. 697–710, 2002.

[tmf18] tmforum. Information Framework (SID), 2018. `https://www.tmforum.org/information-framework-sid/`, accessed: 20/01/2018.

[va17] van der Meulen, P.: Step-up Authentication as-a Service. In: TNC17 - The Art of Creative Networking. 2017.

List of Abbreviations

AuthN	Authentication
AuthN P/S/T/F	AuthN protocols, standards, technologies and frameworks
AuthNI	Authentication Information
AuthZ	Authorization
MFA	Multi Factor Authentication
MNM	Munich Network Management
MSM	MNM Service Model
OIDC	OpenID Connect
OIDC OP	OpenID Provider
OIDC RP	OIDC Relying Party
RAS	Real-World Authentication Subject
SAML	Security Assertion Markup Language
SAML IdP	SAML Identity Provider
SAML SP	SAML Service Provider
TAAS	Trustworthy Authentication Information Administration Service
TAC	Trustworthy Authentication Information Consumer
TAP	Trustworthy Authentication Information Provider
TAPS	Trustworthy Authentication Information Provisioning Service
TAS	Trustworthy Authentication Subject
TAuthNI	Trustworthy Authentication Information
TRRS	Trustworthy Authentication Information relying subject-service
U2F	Universal Second Factor
UASM	Universal Authentication Service Model

Müller, Neumair, Reiser, Dreo Rodosek (Hrsg.): 11. DFN-Forum Kommunikationstechnologien,
Lecture Notes in Informatics (LNI), Gesellschaft für Informatik, Bonn 2018 93

Teaching network softwarization with SDN Cockpit: An open ecosystem for students, network administrators and others

Robert Bauer, Hauke Heseding, Addis Dittebrandt, Martina Zitterbart[1]

Abstract: This paper introduces SDN Cockpit, an easy-to-use and open ecosystem for teaching network softwarization based on mininet and the Ryu controller. The ecosystem allows candidates to gain hands-on-experience with SDN in prefabricated scenarios without having to deal with potentially complex details such as traffic generation. It provides useful tooling for instructors and automated evaluation for assignments. The paper discusses the design goals, the architecture and the workflow of the ecosystem. First experiments with SDN Cockpit show that the approach can improve the motivation and the learning experience of the candidates.

Keywords: Network Softwarization; SDN Cockpit; Teaching

1 Introduction

Network softwarization describes a recent trend to enable flexibility in the network through logically centralized software-based control and virtualization. Software-defined Networking (SDN) is one important key technology in this domain. SDN decouples the control plane and the data plane from each other and allows remote programmability of forwarding devices. It attracted significant attention since it was initiated, but many of the underlying ideas can be traced further back (up to the early days of the internet [FRZ14]).
SDN comes with a number of enticing promises: rapid development of new features, fast time-to-market, purchase and maintenance of hardware and software independently of one another to reduced CAPEX/OPEX, and many others. It is thus not surprising that SDN stepped into datacenters or mobile backbone networks, with no intention to leave anytime soon. Consequently, we believe that at least the core ideas behind SDN will demand their fair share in future networking classrooms.

One thing that is particularly intriguing about SDN: it is an excellent topic for teaching network fundamentals, primarily because it is very easy to get your hands dirty and gain practical experience – especially compared to many other technologies in the networking domain. Understanding the basic concepts behind SDN is relatively simple due to well defined abstractions and a clean three-layered architecture. And a wide variety of powerful tools for rapid prototyping such as mininet [LHM10] exist to assist with practical realizations.

[1] Karlsruhe Institute of Technology, Institute of Telematics, Karlsruhe
robert.bauer@kit.edu, heseding@kit.edu, addis.dittebrandt@student.kit.edu, zitterbart@kit.edu

This holds for a variety of "candidates" to be teached, including network administrators in computer and data centers as well as students in basic and advanced networking classes.

In the past three years, we supervised several SDN-related classes with practical SDN assignments and noticed that a large part of the participants developed a sound understanding of the topic within a tight timeframe, just by experimenting with the technology (which is good news). Approximately three-quarters of the participants were capable of autonomously setting up basic SDN scenarios (learning switch, simple routing, simple load-balancing) with existing tool chains, e.g., based on mininet and the Ryu controller.

However, we also got reports from participants who did not finish the (partly voluntary) assignments. Especially participants with little programming experience and/or affinity tend to give up early. In most cases, we were able to identify one or multiple of the following three obstacles as root cause for the inability to finish an assignment: 1) the perceived complexity of several interacting tools (network emulator, traffic generator, SDN controller), 2) inability to find/identify required pieces of information to correctly utilize the tools and 3) the absence of a clear success metric (do I have the correct solution?). Based on this observation, we argue that there is a demand for an SDN ecosystem that focusses not only on rapid prototyping but also on easy accessibility and the capability to draw the attention to the essential core of SDN. Unfortunately, most existing tools do not focus on these objectives.

In this paper, we first introduce our vision of an open and easy-to-use ecosystem for teaching network softwarization and discuss important design goals. After that, we introduce our prototype for such an ecosystem called SDN Cockpit. The tool is publicly available on GitHub [SD18]. We outline the general architecture of the ecosystem, its workflow and present first insights from practical use.

2 Towards an ecosystem for teaching network softwarization

As outlined in the introduction, we envision an open and easy-to-use ecosystem that allows it to teach the ideas of network softwarization – with a current focus on SDN – to a broad audience. We assume partially supervised self-teaching, i.e., the users of the ecosystem work with a prepared set of assignments following a "learning by doing" approach. This is partially supervised, because the assignments have to be prepared somehow and the ecosystem itself takes the role of a supervisor. We further assume that many learning objectives in the context of network softwarization can be achieved by experimenting with the control plane, or, to be more precise, by developing control plane applications[2].

From a high level perspective, we consider two types of stakeholders (see also Figure 1): *Instructors* capable of setting up assignments for a given learning objective (e.g., understand

[2] Note that there are also concepts of network softwarization that require a different angle, e.g., middlebox virtualization or data plane optimizations. Such concepts are currently not supported by SDN Cockpit.

reactive flow programming, understand multipath routing, . . .) and *candidates* – say students or network administrators – that use these assignments. The ecosystem then consists of two major parts: a frontend part which must be visible to the candidates and a separate backend part which might or might not be visible. There is a human machine interface between the candidates and the frontend (to work on assignments) and another interface between the instructor and the backend of the ecosystem (e.g., to include new assignments). We provide further details when discussing the general architecture of our prototype in Section 3.1.
We will now go through several design goals for the ecosystem in greater detail, followed by a short discussion for two especially important design decisions.

2.1 Design Goals

Easy accessibility: The steps or preparations necessary from a candidate to get started with the first assignment should be as low as possible, which includes initial access to the ecosystem, access to the frontend and selection of an assignment. If these steps consume too much time or the setup procedure is too complex – e.g., because several dependencies have to be installed –, the initial hurdle for using the ecosystem might be too high. In fact, we see only three possible deployment schemes that can fit these criteria: 1) Remote access to the frontend, e.g., in the form of a web interface, 2) provisioning in the form of a consolidated stand-alone program (single dependency) or 3) provisioning in a container with the help of virtualization. We also use the term accessibility to refer to the general effort that is required by a candidate to get started with an assignment. In the best case situation, the candidate can edit a solution and gets immediate feedback to the changes.

Easy transition: The ecosystem should be designed in such a way, that an easy transition to real technology is possible without too much effort. That means, that the candidate can "leave" the ecosystem to continue experimentation without potential limitations and restrictions of the ecosystem (enforced by other design goals). In the case of SDN, this would mean that the applications written in an assignment can be executed using only the underlying SDN controller (without the ecosystem!) or that the traffic can be replicated by stand-alone scripts or generators.

Auditability of solutions: Providing a clear success metric to candidates is critical. The ecosystem should thus not only provide the assignments, it should also trace the current state of an assignment and provide positive and/or negative feedback to candidates. The latter aspects have to be fully automated to avoid having an instructor in the loop. Such a design has several benefits: 1) the candidate has a clear objective and is able to autonomously (and transparently!) trace the progress of the current assignment, 2) a continuous feedback loop between the candidate and the ecosystem can assist the learning process and 3) it enables auditing capabilities for instructors if required.

Flexible scenarios: To support a broad range of different learning objectives, the ecosystem has to support different topologies and different traffic profiles. A traffic profile is an

abstract formulation of packets that are exchanged on top of a topology. In this context, we refer to an *integrated scenario* as the combination of a topology and a traffic profile generated specifically for this topology. Interesting integrated scenarios for SDN include load balancing scenarios (flow demand has to be scheduled to multiple links) or inter-domain routing scenarios (policies have to be applied to flows). Note that simple traffic generators like iperf might be not sufficient here, e.g., in the case of customized source and destination addresses.

Other design goals: A clean and modular design of the ecosystem helps with extendability. External dependencies (e.g., controllers) should be integrated in a lightweight fashion to support easy transition. Platform independence is also required so that candidates can use their habitual operating system. Last but not least, we are convinced that an ecosystem for teaching network softwarization is more likely to be successful if it openly approaches the community, which includes free access to the source code and the assignments.

2.2 Important Design Decisions

One major design issue is related to the technical core of the ecosystem, i.e., the question whether to use *emulation* or *simulation*. The latter has the big advantage that the ecosystem would be more controlled, deterministic and independent of the underlying hardware (better auditability). On the downside, existing simulators with SDN support such as ns-3 or OMNeT++ are restricted to an internal controller API [Iv16] which might suffer from a scarcity of examples and – more important – prevents easy transition. Emulation, on the other hand, is directly based on real networking stacks and compatible with any SDN controller which results in an authentic experience for the candidates (better transition). Because easy transition is one of the central design goals and both approaches can be designed for easy accessibility, we decided to use emulation based on the mininet tool [LHM10]. Another important design issue is the choice of the controller architecture. While production-grade controllers such as ONOS and OpenDaylight offer high availability, resiliency and performance, the complexity of the application programming interface of Python-based controllers like Ryu is much lower. In this case, easy accessibility outweighs easy transition so that we chose Ryu [Ry18] as the controller that best fits into our ecosystem.

3 SDN Cockpit

SDN Cockpit was designed following the goals outlined in Section 2.1. It is based on the general idea, that the candidates are confronted with the lowest possible amount of distraction. In fact, they have to deal with only two simple interfaces while doing an assignment: a text editor they are familiar with (to edit the solution) and an aggregated view summarizing all currently required pieces of information. Everything else is hidden by the ecosystem. In the following, we discuss the architecture, the workflow and first insights from practical use.

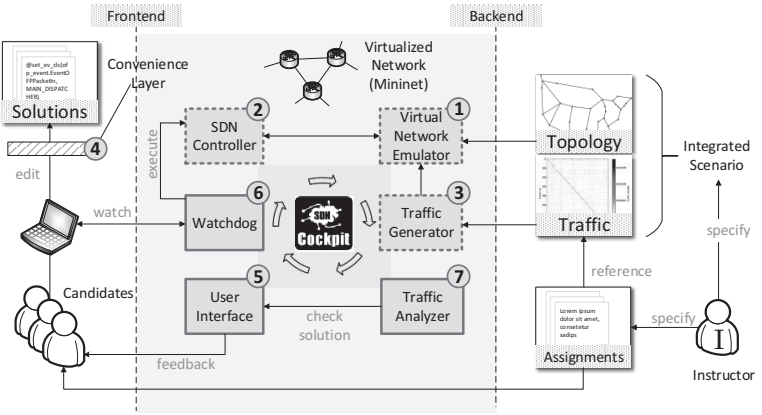


Fig. 1: Architecture of SDN Cockpit

3.1 Architecture

The overall architecture of the ecosystem is given in Figure 1. SDN Cockpit consists of seven components and two interfaces, one for candidates (frontend) and one for instructors (backend). The processes within the grey area between the two interfaces form the ecosystem. The virtual network emulator, the SDN controller, and the traffic generator are based on unmodified open-source projects (① to ③). The convenience layer, the user interface, the watchdog, and the traffic analyzer are new components introduced by SDN Cockpit (④ to ⑦) . The components are woven together with a set of Python scripts represented by the black icon in the middle. The whole ecosystem is usually bundled within a virtual machine and can be deployed with a single call to the command line using vagrant [Va18]. In the following, we go through the individual components of the architecture and describe their role in the ecosystem.

Virtual Network Emulator : The virtual network emulator – mininet in our case – represents the *data plane* of the ecosystem, i.e., it provides a set of SDN-capable switches connected according to a predefined topology. Despite being virtual, the switches are required to possess the same functional characteristics as standard OpenFlow-compliant switches in order to ensure an authentic experience and support transition. Furthermore, mininet can integrate virtual hosts into the network and virtualization techniques provide each virtual host with its own isolated execution environment. The emulator is configured by the instructor, which includes a topology and its parameterization (bandwidth and latency constraints). In order to support auditability, the emulator provides feedback on the success or failure of network-targeted operations to the candidate via the user interface (not shown in the figure).

SDN Controller: The controller represents the *control plane* of the ecosystem. The applications (= solutions) provided by the candidates are executed here. The controller itself is unmodified, but the SDN Cockpit ecosystem takes care of several basic tasks usually required for using the controller properly. The ecosystem might also rely on partial pre-configuration. For example, connectivity on lower network layers can be provided beforehand when this networking aspect is not of concern. A candidate can then focus on the aspects of network programming that are of primary concern in a particular assignment. SDN Cockpit uses the Ryu controller framework due to reasons outlined in Section 2.2.

Traffic Generator: This component is responsible for injecting genuine packets into the network. Automating this process is essential for easy accessibility, because the interaction between traffic generator and network emulator can be very complex, especially for sophisticated scenarios. In SDN Cockpit, we currently use the trafgen tool [ne18] controlled by a set of Python scripts. This solution is highly customizable and allows for a wide range of networking scenarios, including, for example, different protocols, variations in bandwidth utilization, timed emission of packets, and randomized packet contents. Furthermore, in conjunction with the virtual network emulator a variety of communication relations can be modeled, since the emulator can isolate several instances of traffic generators in a given topology from each other.

Convenience Layer: We observed that eliminating undesired complexity can be a key enabler to improve acceptance and usability for instructors and learning effectiveness for candidates. Hence, we introduce a convenience layer to simplify interaction with the ecosystem. This simplification is twofold:

> a) The programming interface for the candidates is reduced to the functionality that is required to achieve a desired goal. The interface against which the candidates implement their solutions is a set of functions that are essentially derived from the programming interface of the SDN controller. However, the complex syntax, object structure and unnecessary controller internals remain hidden.
>
> b) Instructors are provided with a convenient set of configuration options that can be used to easily create new assignments. The current configuration interface provides options to define traffic flows that should occur during an assignment, e.g., static flows (ensure accurate reproducibility) or randomized flows to emulate dynamic network behavior. These options are presented in a comprehensive configuration format so that an instructor does not need to be aware of the different configuration options of the various tools (e.g., complex traffic generator configuration files or parameters of a network emulator).

User Interface: While the convenience layer accepts input from the candidates, the user interface is the primary source of feedback. A screenshot of the interface is given in Figure 2. It is structured to simultaneously display output from the controller as well as from the traffic generator and analyzer. Through this aggregated view it is possible for a candidate

to comprehend how the network reacts to the current solution. The controller can provide feedback whether or not a control action targeted at the network was successfully executed. Furthermore, the forwarding behavior in the network itself can be observed from the output of the traffic analyzer. A correct solution is immediately indicated by signaling success to the candidate. Faulty network behavior, on the other hand, will result in inaccurate packet counts, either through packets arriving at unintended destinations or not being forwarded at all. In any case, the candidate is presented with a metric that represents the degree of success, i.e., the number of correctly forwarded packets. This feedback can be used to incrementally improve the solution.

Watchdog: The watchdog continuously monitors changes to the solution made by the candidate. Once a change has been detected, it is the watchdogs' responsibility to restart certain components of the ecosystem automatically. This involves resetting the state of the entire network, i.e., the SDN controller, the virtual network emulator, as well as traffic generation and analysis. The automization of this process relieves the candidate from the tedious task of managing the network state himself to ensure undistorted results. The watchdog is essential to provide easy accessibility and auditability of solutions.

Traffic Analyzer: The traffic analyzer component monitors the packets that are sent and received at communication endpoints throughout the network. As soon as the traffic generation process for one of the integrated scenarios is complete, it evaluates the correct forwarding behavior based on the expected packet count for each endpoint. These counts are determined dynamically through monitoring of the traffic during the generation phase so that even randomized packets will produce accurate results. Randomization of traffic can be used to model the unpredictable behavior of real networks. Hence it serves to facilitate our design goal of an easy transition. Special care must be taken during the collection of packets, since management traffic generated by the virtual network emulator itself can

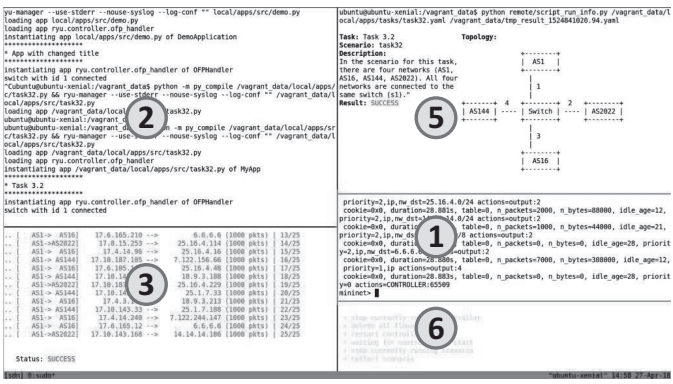


Fig. 2: SDN Cockpit user interface with SDN controller (2), traffic generator and analyzer (3), assignment info (5), mininet emulator (1) and watchdog (6)

interfere with packet count accuracy. The traffic analyzer provides immediate feedback to a candidate once traffic generation and analysis have been completed. Hence, the candidate can evaluate the feasibility of his solution in a timely manner.

3.2 Workflow Example

This section presents an example workflow showing all the steps of using the ecosystem from assignment creation by the instructor to submission of the solution by the candidate. SDN Cockpit is distributed either via git or as a standalone vagrant image which can be downloaded from an online repository. Installation of the tool is performed by executing `vagrant up` either inside the git directory or with the image path as an argument inside a new directory. In the following, we assume that the candidate has installed the environment already.

The instructor first has to create an assignment geared towards a specific learning goal. We choose ”policy based routing“ as an example. The assignment is to route traffic of an Autonomous System (AS) based on a static set of policies. It consists of an assignment text and an integrated scenario specifically designed for the assignment so that the solutions can be tested. The assignment can be delivered automatically by the ecosystem (see ⑤ in Figure 2). The candidate launches SDN Cockpit by executing `vagrant ssh` followed by `./run.sh`. The frontend is divided into 5 panes: a pane showing the output of the ryu controller ②, a pane illustrating traffic generation and traffic analysis ③, a pane for assignment information ⑤, a pane for interacting with mininet ① and a pane for watchdog activity ⑥. The assignment information pane shows essential information such as a textual description (e.g., which AS should be prioritized), a visualization of the topology and a mapping of network subnets to ASes and switch ports. The latter is required for flow programming.

The candidate proceeds with solving the assignment by opening the corresponding solution file externally with a text editor. It contains an initial ryu application skeleton which is not a valid solution to the assignment (yet). The skeleton integrates functions of the convenience layer tailored to the assignment. In this case, it would include helper functions for easy flow programming. The candidate then writes an initial solution which incorrectly programs the routing policies. Upon saving, the watchdog ⑥ shows that the saved file has been registered and a test run is executed. Pane ② can be consulted for the controller output. It can be used to show abnormal behavior of the application through logs and exceptions. Once the traffic generator terminates, the analyzer will check how many packets have been received by which AS. As the solution does not yet work properly, the traffic analyzer detects this violation through deviating packet counts and presents a failed test case to the candidate. To further dissect the problem, the candidate can use the mininet CLI ① to manually send traffic and observe the reaction of the application in pane ② or by dumping flow statistics through the mininet CLI. This loop is repeated until the solution passes the tests. Small incremental steps towards a valid solution are encouraged with this approach as short test runs are executed immediately upon saving which results in fast feedback.

Tab. 1: Success rate for two (slightly different) sets of assignments from 2017 without SDN Cockpit and 2018 with SDN Cockpit. Failure indicates that a candidate missed the learning objective.

without SDN Cockpit 2017, n=13			
	Submitted	Success	Failure
Easy	100 %	100 %	0 %
Medium	92 %	69 %	23 %
Difficult	0 %	/	/

with SDN Cockpit 2018, n=17			
	Submitted	Success	Failure
Easy	100 %	94 %	6 %
Medium	94 %	94 %	0 %
Difficult	65 %	53 %	12 %

3.3 Experience Report

So far, we have successfully used the SDN Cockpit ecosystem for two different activities: a completely voluntary assignment as part of an advanced networking class (10+ candidates, not discussed here further) and an obligatory assignment for a practical course with 17 candidates. The latter one included three tasks with increasing difficulty (easy, medium and difficult). To have some kind of comparison, we take the results from an earlier SDN-assignment with 13 participants that was conducted in early 2017 with a very similar setup but without SDN Cockpit: same background, same time frame, same technology, similar size/difficulty of the tasks. The students manually installed the required software (Ryu, mininet) and used ping/iperf for basic traffic generation. In both cases, the third and most difficult task could be submitted on a voluntary basis. Table 1 shows a high level analysis of the results.

In both cases, the majority of the candidates were able to meet the learning objective for the easiest task (basic flow programming). For the task with medium difficulty (still flow programming but in a more complex scenario), there seems to be an improvement with regard to the percentage of candidates that were able to meet the learning objective when working with SDN Cockpit – 23% failure rate for 2017 compared to 0% failure rate in 2018. The most important outcome, however, is the percentage of candidates that have worked on the voluntary task. While we received no solution for this task in 2017, 65% of the candidates provided a solution in 2018. Because of the low sample size and the fact that the two groups worked on slightly different tasks, the numbers in Table 1 must be treated with considerable caution. However, at least the overall trend is in line with our personal experience from working with the tool and the participants. In conclusion, we believe that an easy-to-use ecosystem can improve the motivation and the learning experience of the candidates. The SDN Cockpit approach might be a first step in this direction.

4 Conclusion and Future Work

This paper introduced SDN Cockpit as a novel ecosystem for teaching network softwarization. We put easy accessibility and the possibility to transition to real technology as our top design goals and build the ecosystem around them. We gathered first practical experience

with the ecosystem in a university context with two different classes and currently plan to extend this in the context of the bwNET100G+ project [bw18] to also include network and system administrators.

While we are pretty pleased with the initial results of SDN Cockpit, there are several aspects that we want to address in the future. First, there is still room for improvement with regard to accessibility. It requires some time to get familiar with the process of assignment selection and the tmux-based user interface. In addition, candidates should be allowed to tune certain aspects of the integrated scenario (topology, traffic) to further improve the learning experience – which is currently not supported. Second, the automatic evaluation of more complex scenarios is difficult to set up. And third, the current deployment scheme takes a non-negligible amount of time and hardware resources. We are currently working on a newer version of SDN Cockpit where the frontend is implemented in the browser and the backend runs in a light-weight container.

Despite these limitations, we are confident that practical experimentation with SDN Cockpit is not only useful for understanding SDN, it can also be a promising ecosystem for getting insights into other related areas, e.g., inter-domain routing, load balancing algorithms or optimization theory. With respect to the latter, it might even be applicable for purposes outside of the domain of communication networks.

References

[bw18] bwNET100G+: Research and innovative services for a flexible 100G-network in Baden-Wuerttemberg. https://www.bwnet100g.de/, 2018. (Accessed on 02/01/2018).

[FRZ14] Feamster, Nick; Rexford, Jennifer; Zegura, Ellen W.: The road to SDN: an intellectual history of programmable networks. Computer Communication Review, 44:87–98, 2014.

[Iv16] Ivey, Jared; Yang, Hemin; Zhang, Chuanji; Riley, George: Comparing a Scalable SDN Simulation Framework Built on Ns-3 and DCE with Existing SDN Simulators and Emulators. In: Proceedings of the 2016 ACM SIGSIM Conference on Principles of Advanced Discrete Simulation. SIGSIM-PADS '16, ACM, New York, NY, USA, pp. 153–164, 2016.

[LHM10] Lantz, Bob; Heller, Brandon; McKeown, Nick: A Network in a Laptop: Rapid Prototyping for Software-defined Networks. In: Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks. Hotnets-IX, ACM, New York, NY, USA, pp. 19:1–19:6, 2010.

[ne18] netsniff-ng toolkit. http://netsniff-ng.org/, 2018. (Accessed on 02/01/2018).

[Ry18] Ryu SDN Framework. https://osrg.github.io/ryu/, 2018. (Accessed on 02/01/2018).

[SD18] SDN Cockpit. https://github.com/kit-tm/sdn-cockpit, 2018. (Accessed on 27/07/2018).

[Va18] Vagrant by HashiCorp. https://www.vagrantup.com/, 2018. (Accessed on 02/01/2018).

Müller, Neumair, Reiser, Dreo Rodosek (Hrsg.): 11. DFN-Forum Kommunikationstechnologien,
Lecture Notes in Informatics (LNI), Gesellschaft für Informatik, Bonn 2018 103

Evaluation von VIRL, GNS3 und Mininet als Virtual Network Testbeds in der Hochschullehre

Christoph Seifert[1], Sven Reißmann[2], Sebastian Rieger[1], Christian Pape[1]

Abstract: In den letzten Jahren wurden verschiedene Frameworks zur Implementierung virtueller Computernetzwerkumgebungen entwickelt. Typischerweise können diese in Simulations- und Emulationsansätze kategorisiert werden. Letztere zeichnen sich dabei häufig durch eine im Vergleich zu Simulationsansätzen höhere Praxisnähe und einen realitätsnahen Funktionsumfang aus. Die Einrichtung und Wartung dieser Tools, z.B. in Laborumgebungen von Universitäten, ist jedoch komplex. In diesem Papier werden moderne Netzwerk-Emulations-Tools für Computer-Networking-Kurse und Forschungsprojekte im Netzwerklabor (NetLab) der Hochschule Fulda evaluiert. In erster Linie werden GNS3, VIRL und Mininet ausgewertet, die derzeit in Lehrveranstaltungen in Bachelor- und Masterstudiengängen im NetLab eingesetzt werden. Besonderes Augenmerk wird auf die Skalierbarkeit für große Studierendenzahlen, geringen administrativen Aufwand und Kosten, hohe Praxisnähe der unterstützten Laborübungen und entsprechende didaktische Anforderungen gelegt.

Keywords: Netzvirtualisierung, Hochschullehre, VIRL, GNS3, Mininet

1 Einleitung

Die Paradigmen im Bereich der Lehre haben sich im Verlauf der vergangenen Jahre stark verändert. E-Learning und Blended Learning erlauben zeit- und ortsungebundenes Lernen und relativieren teils sogar die Erfordernis der Präsenz von Studierenden in Vorlesungen. Lehrveranstaltungen in Laboren im Allgemeinen, insbesondere in Netzwerklaboren, lassen sich jedoch häufig nicht ohne weiteres auf diese neuartigen Lehrmethoden übertragen. Beispielsweise ist für Trainings an physischen Geräten oder zur Stärkung von Teamwork häufig eine körperliche Präsenz der Studierenden in Praktika sinnvoll oder gar notwendig. Während Vorlesungen in der Regel 90 Minuten dauern, erfordern Laborveranstaltungen zusätzliche Vor- und Nachbereitung. Zum Beispiel müssen in Netzwerklaboren für Übungen zu Beginn jeder Veranstaltung die erforderlichen Hard- und Software-Umgebungen aufgebaut und konfiguriert werden. Dies bedeutet insbesondere bei steigenden Teilnehmerzahlen in Laborkursen einen erhöhten Aufwand. So bleiben von den regulären 90 Minuten häufig nur 60 Minuten für die Durchführung der eigentlichen Praktika.

Virtuelle Labore können dabei helfen eben diese Probleme zu adressieren, indem sie einerseits schnell auf einen definierten Zustand vorbereitet werden können und andererseits

[1] Hochschule Fulda, Angewandte Informatik, [vorname.nachname]@informatik.hs-fulda.de
[2] Hochschule Fulda, Rechenzentrum, sven.reissmann@rz.hs-fulda.de

erlauben, den aktuellen Zustand zu speichern, um später genau an diesem Punkt fortzufahren. Virtuelle Labore können durch Simulatoren oder Emulatoren realisiert werden. Da die Simulation jedoch oft nicht vollständig dem Verhalten von realen Geräten entspricht, wie [Ha12] und [YM17] zeigen, bieten Emulatoren häufig die bessere Wahl. Abbildung 1 zeigt einige Charakteristiken von Emulatoren, Simulatoren und Testbeds.

	Simulatoren	Testbeds		Emulatoren
		Shared	Custom	
Realitätsnahe Funktionen		✔	✔	✔
Realistisches Timing	✔	✔	✔	(einge-schränkt)
Realistischer Traffic		✔	✔	✔
Flexibilität der Topologien	✔	(einge-schränkt)		✔
Einfache Nachvollziehbarkeit	✔	✔		✔
Geringe Kosten	✔			✔

Abb. 1: Lösungen für reproduzierbare Netzexperimente aus [Ha12].

Im Netzwerklabor (NetLab) des Fachbereichs Angewandte Informatik der Hochschule Fulda werden praktische Übungen für die Lehrinhalte des Cisco Networking Academy Programms mit dem Simulationstool Packet Tracer [18g] unterstützt. Diese Software kann für die im Rahmen dieses Programms angebotenen CCNA-Kurse alle erforderlichen Aspekte und Funktionen virtuell abbilden. Packet Tracer erlaubt es jedoch nicht Funktionalitäten zu implementieren, die für fortgeschrittenere CCNA-Sicherheits- oder CCNP-Kurse erforderlich sind. Darüber hinaus ist die Praxisnähe der Software aufgrund fehlender Funktionen (z.B. Anbindung der Simulation an das Internet) und fehlendem realitätsnahem Netzwerk-Traffic bedingt durch den Simulationsansatz eingeschränkt. Aus diesen Gründen sind sowohl Packet Tracer als auch andere Simulatoren kein Bestandteil der Analyse in diesem Paper. Die meisten Emulatoren bzw. virtuellen Netzwerk-Umgebungen lassen sich auf Plattformen wie Linux, Windows und auch Mac OS realisieren. Auch eine Bereitstellung auf virtuellen Plattformen, z.B. VMware-basierte Server-Lösungen, wird unterstützt. In dieser Ausarbeitung konzentrieren wir uns auf Werkzeuge, die die meisten der im Abschnitt 3 genannten Kriterien für den Einsatz im Umfeld des NetLab erfüllen. Der Schwerpunkt wird hierbei auf das Virtual Internet Routing Lab (VIRL)[18i] von Cisco sowie die Open Source Lösungen GNS3 [18d] und Mininet [18e] gelegt, die alle im NetLab für unterschiedliche Einsatzzwecke aktiv eingesetzt werden. VIRL ist mittlerweile nur noch als Einzelplatzlizenz nutzbar. Die auf der gleichen Basis aufsetzende Mehrplatzlösung Cisco Modeling Lab (CML) [18a] wird aufgrund ihrer hohen Lizenzkosten nur der Vollständigkeit halber erwähnt.

2 Verwandte Arbeiten

Moderne IT-Infrastrukturen basieren auf komplexen Netztopologien, die die Grundlage für Skalierbarkeit, Redundanz und Hochverfügbarkeit von IT-Services bilden. Es ist

offensichtlich, dass die Vermittlung theoretischen Wissens alleine nicht genug ist, um Studierenden der Informatik eine gute Grundlage zum Verständnis solcher Topologien zu bieten. Daher sind die Simulation und Emulation von Netzwerktopologien an Hochschulen Gegenstand aktueller Forschung und kontinuierlicher Weiterentwicklung. Ein Vergleich einer rein VIRL-basierten Lernumgebung mit physischen Setups (CCNA-Pods) und der Simulation mit Cisco Packet Tracer wird in [Ta16] vorgestellt. In [Ob14] wird der Einsatz von VIRL für Forschung und Lehre skizziert und bewertet. Die früher im NetLab vorrangig eingesetzte VIRL-Umgebung und deren Skalierbarkeit wurden bereits in [Ri17] und [SRP17] vorgestellt. Ein auf einer deklarativen XML-basierten Modellierungssprache basierendes Framework, das erweiterbare und skalierbare Emulationen/Simulationen von großen Netzwerktopologien realisiert, ist in [MK16] dargestellt. Ein Vergleich mit anderen bekannten Simulations- und Emulationswerkzeugen, z.B. ns-3 oder PlanetLab ist ebenfalls enthalten. Netztopologien mithilfe virtueller Router auf Linux-Computern wurden z.B. in [Ba03] emuliert. Der Echtzeit-Netzwerkemulator EmuLab, der für das Testen von Protokollen und des Anwendungsverhaltens in Netzwerktopologien verwendet wird, ist in [KE04] näher beschrieben. [PR16] zeigt einen didaktisch ausgerichteten Ansatz, der darauf abzielt, reale Netzwerkgeräte mit geringen Kosten und geringem Aufwand zu modellieren und zu emulieren. [Ha12] stellt ein Rahmenwerk für eine reproduzierbare container-basierte Emulation von netzbezogenen Experimenten einschließlich Software Defined Networking (SDN) vor. Reproduzierbare Experimente und Forschungsergebnisse im Netzwerkbereich werden zudem in [YM17], [Fl17] und [Ba17] thematisiert.

3 Anforderungen und Auswahlkriterien

Für die Durchführung von Laborübungen mit dem Fokus auf Netzwerktechnologien und -protokolle wurden verschiedene Optionen evaluiert. Diese können, wie in Abschnitt 1 erläutert, typischerweise in Simulatoren, physische oder virtuelle Testbeds und Emulatoren kategorisiert werden. Abbildung 2 veranschaulicht die Einordnung dieser Optionen zwischen Praxis und Theorie. Die im Bild gezeigten Gradienten ergeben sich aus Erfahrungen mit verschiedenen Werkzeugen für physische und virtuelle Laborumgebungen, die im NetLab gewonnen wurden. Physische Testbeds verwenden echte Netzwerkgeräte um Testnetzwerke einzurichten, die entweder isoliert oder in das Netzwerk des Labors integriert und an das Internet angebunden sind. Sie wurden für den Bachelor-Studiengang und das CCNA-Programm, das parallel zum Curriculum angeboten wird, eingesetzt. Um die Flexibilität des Setups zu erhöhen und die Vorbereitungszeit zu reduzieren, wurden virtuelle Testbeds unter Verwendung von VMware Workstation und Arista vEOS verwendet, die jedoch individuell erstellt und bereitgestellt werden mussten.

Emulatoren wie GNS3 [18d], EVE-NG [18b] / UNetLabv2 [18h], VIRL [18i], CML [18a], Mininet [18e] oder eNSP [18c] lassen sich zur automatisierten Erstellung virtueller Netze und Bereitstellung der benötigten Konfigurationen nutzen. Sie ermöglichen dadurch eine signifikante Reduzierung der Vorbereitungszeit. Eine weitere Abstraktion von realen

Netzwerken führt zu Simulatoren (z.B. Packet Tracer [18g], ns-3 [18f]), die zwar die Realität weniger exakt abbilden, dafür aber einen signifikant geringeren Ressourcenbedarf aufweisen, wodurch auch die Realisierung sehr großer Topologien ermöglicht wird. Theoretische Modelle bieten eine hohe Flexibilität, erfüllen aber nicht die Anforderungen bezüglich der Praxisnähe, die im Labor begleitend zur Vorlesung erreicht werden soll.

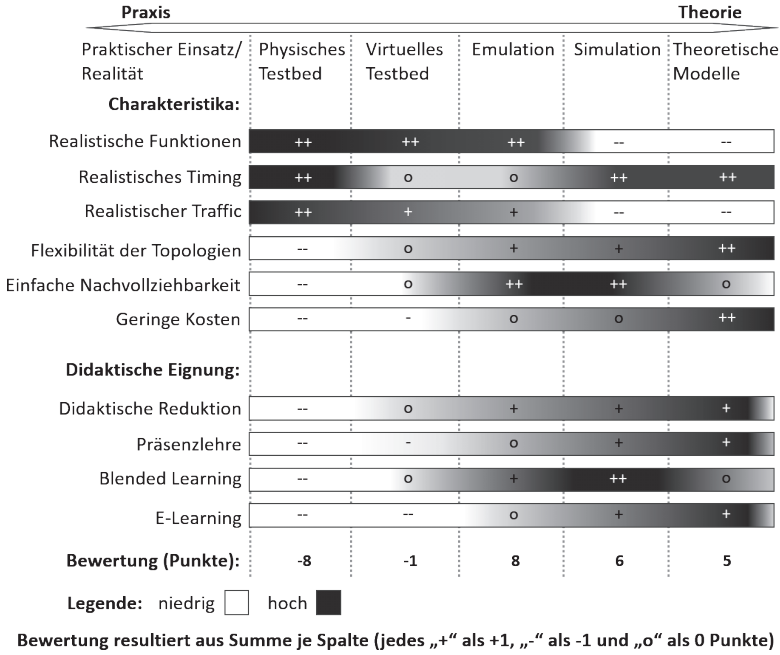


Abb. 2: Klassifizierung unterschiedlicher Ansätze für experimentelle Netzwerkumgebungen.

Zusätzlich zu den in Abbildung 1 aufgeführten Charakteristiken gemäß [Ha12] wurden didaktische Anforderungen an die verschiedenen Ansätze evaluiert, die sich auf die Qualität der Lehrveranstaltungen im NetLab auswirken. Testumgebungen weisen meist komplexe Ansätze auf, die bereits bei der Erstellung des Ausgangszustands eines Versuchs zur Überforderung führen. Durch didaktische Reduktion der Versuchsdurchführung soll der Fokus auf die wesentlichen Lernerfahrungen ausgerichtet werden. Darüber hinaus wurde die Eignung der einzelnen Ansätze für traditionelle und für moderne Lehrmethoden wie E-Learning- und Blended-Learning überprüft. E-Learning und Blended-Learning erfordern einen zeitlich und örtlich unabhängigen Zugang zur Testumgebung und damit gleichzeitig die Möglichkeit, die experimentelle Umgebung leicht und ohne direkte Anleitung verstehen und modifizieren zu können. Integrierte Umgebungen wie sie von Simulatoren und Emulatoren zur Verfügung gestellt werden, besitzen damit eine hohe Eignung für die Realisierung virtueller Netzwerkumgebungen. Sie bieten eine hohe Praxisnähe, realistischen Funktionsumfang, ermöglichen das Pausieren und Abspeichern des aktuellen Zustands

und reduzieren den Administrationsaufwand erheblich. Die in Abbildung 2 dargestellte Bewertung ergibt sich aus Erfahrungen basierend auf regelmäßigen Laborveranstaltungen im NetLab mit einer typischen Dauer von 90 Minuten. Vor allem wegen des manuellen Setups, der damit verbundenen Vorbereitungszeit und des Wartungsaufwands erhalten virtuelle Testbeds -1 Punkt. Folglich erhalten physische Testbeds, die zwar noch näher an der Realität liegen, aber entsprechend aufwändiger zu konfigurieren und ungeeignet für E-Learning sind, insgesamt -8 Punkte. Solche Testbeds werden im NetLab zwar ebenfalls aktiv eingesetzt, um komplexe Szenarien abzubilden, die besonders hohe Praxisnähe erfordern, jedoch bieten andere Implementierungsmöglichkeiten eine höhere Flexibilität und Effizienz. Besonderes Augenmerk wird auf den Einsatz von Emulatoren gelegt, die gewissermaßen einen Mittelweg bieten, um die Vorteile aus Theorie und Praxis zu vereinen. Sie stellen jedoch neben den genannten Vorteilen hohe Anforderungen an die benötigten Ressourcen, da typischerweise Betriebssysteme bzw. Images von realer Netzwerk-Hardware in Form von virtuellen Maschinen mit hohem Arbeitsspeicherbedarf ausgeführt werden. Die mögliche Größe der emulierten Netze verhält sich somit - im Gegensatz zu Simulatoren - linear zu den verfügbaren Rechen- und Speicherressourcen.

Zur Bewertung der jeweiligen Lösungen wurden nachfolgend beschriebenen Kriterien herangezogen herangezogen (vgl. Abbildung 3). Einen wichtigen Auswahlfaktor für den Betrieb im NetLab stellen die Lizenzkosten dar, die mit einer entsprechend hohen Gewichtung in die Bewertung eingehen. Kriterien wie ein zentrales Management der Umgebung, hohe Kompatibilität (z.B. bei der Verwendung verschiedener Betriebssysteme als Hosts für die Emulation) und die Zugangsmöglichkeiten im Hinblick auf E-Learning wurden mit etwas geringerem Gewicht in die Bewertung einbezogen. Mit einer etwas höheren Wertung wurde die Möglichkeit der Integration von Images realer Netzwerkgeräte einbezogen, um eine hohe Praxisnähe von experimentellen Labor-Umgebungen zu ermöglichen. Dies ist im NetLab aufgrund der praktischen Verwendung in Cisco-basierten Kurse (z.B. CCNA, CCNP), aber auch um eine hohe Realitätsnähe durch Verwendung echter Switch- und Router-Betriebssysteme verschiedener Hersteller zu erreichen, ein wichtiges Kriterium. Weitere Kriterien stellen die Ressourcenanforderungen und Skalierbarkeit der Lösung dar, die bei typischerweise 20 Teilnehmern in den Kursen die Vor- und Nachbereitungszeit von Laborveranstaltungen sowie die Performance innerhalb der Testumgebung maßgeblich bestimmen. Bei einem zentralen Betrieb der Emulationslösung auf Servern des NetLab, ist eine automatisierte Lastverteilung der von Studierenden gestarteten virtuellen Ressourcen und Testbeds (z.B. über einen verteilten Cluster) wünschenswert. Eine Reihe technischer Auswahlkriterien mit direktem Einfluss auf die didaktische Flexibilität der Lösungen wurde ebenfalls in die Bewertung einbezogen. Dazu gehört zunächst die Anbindung der Umgebung an ein reales Netz wie das physische Labornetz im NetLab und das Internet, z.B. um die Kursteilnehmer in die Lage zu versetzen, reale Werkzeuge (z.B. arp, ping, traceroute, Wireshark) in ihren Experimenten wie gewohnt und mit realen Zielen und realistischem Datenverkehr verwenden zu können. Gleichzeitig spielt die Anbindung an das Internet eine wichtige Rolle um den Zugang zu den emulierten Netzwerken über VPN zu ermöglichen und damit die Anforderungen im Bezug auf E-Learning zu erfüllen.

		mininet		VIRL		CML		GNS3 (V2.1)	
Kriterium	**Gewicht 1-7**	**Bemerkung**	**#**	**Bemerkung**	**#**	**Bemerkung**	**#**	**Bemerkung**	**#**
Administrativer Aufwand									
- Lizenz (Kosten)	7	Frei (Open Source)	10	199€ für 20 Cisco Nodes p.a. (nicht skalierbar, Einzelplatz-Lizenz)	4	Kostenintensiv (> $2.500 p. a. {für 15+10 Nodes}), skalierbar (zus. Kosten)	0	Frei (Open Source, GPLv3)	10
- Zentrales Management	3	Isoliert (Einzelplatz), aber per Skript automatisierbar	7	Cluster mit mehreren Compute Nodes (OpenStack)	10	Cluster mit mehreren Compute Nodes (OpenStack)	10	Mehrere aber nicht zentral verwaltete Server möglich	3
- Kompatibilität und Zugänglichkeit	2	verfügbar für Linux (Win, Mac als VM)	7	VMMaestro Client und Server VMs für Linux, Win, Mac, vSphere	8	VMMaestro Client und Server VMs für Linux, Win, Mac, vSphere	8	Verfügbar für Linux, Win, Mac, auch als vorkonfigurierte VM	7
- Custom Images für Netzhardware	4	Kein Support für Images realer Netzhardware	0	Erweiterbar (nur für Cisco Nodes beschränkt), Third-Party Images verfügbar	5	Erweiterbar (nur für Cisco Nodes beschränkt), Third-Party Images verfügbar	5	Images für Vielzahl Router, Switches (Cisco Images nicht enthalten)	7
- Load Balancing	2	Nur manuell	5	Ja	10	Ja	10	Nur manuell	5
- Erforderliche Ressourcen	1	Gering (LXC Container)	10	Device-abhängig	6	Device-abhängig	6	Device-abhängig	8
Didaktische Anforderungen									
- Anbindung an reales Netz	4	Ja	8	Ja	8	Ja	8	Ja (NAT über GUI, Bridge manuell)	6
- Mehrere Benutzer und Sitzungen	5	Nur manuell	5	Nein (offiziell nur single user)	8	Ja	10	Ein Projekt pro User, aber Zugriff mehrerer Clients möglich	6
- Konsolenverbindung	2	Beschränkte LXC Konsole	6	Jeder Benutzer hat eigene Konsole	6	Jeder Benutzer hat eigene Konsole	6	Gemeinsame Konsole (User)	10
- QoS Emulation auf Links	1	Ja	10	Ja (nur Delay, Jitter, Loss)	8	Ja (nur Delay, Jitter, Loss)	8	Ja (nur Delay, Jitter, Loss)	8
- VPN Zugang zu virtuellen Netzen	1	Manuell einrichtbar	5	Ja	10	Ja	10	über VPN Server in virtuellem Netz	5
Gewichtete Bewertung			209		222		204		226

Abb. 3: Evaluierung im NetLab eingesetzter Netz-Emulatoren basierend auf Erfahrungswerten.

Studierende können sich so mit ihrem Rechner direkt in die von ihnen gestarteten virtuellen Testnetze verbinden. Um die Praxisnähe von virtuellen Netzen zusätzlich zu steigern, kann die Emulation von QoS-Metriken (z.B. Packet Loss, Delay, Jitter) auf Verbindungen in den virtuellen Netzen sinnvoll sein. Die Eignung der jeweiligen Lösung für diese Forderung spiegelt sich im Bewertungskriterium *QoS Emulation auf Links* wieder. Schließlich wurden auch Möglichkeiten der Kollaboration in Bezug auf die didaktischen Anforderungen

bewertet. Diese sind in unserem Setup essentiell, da sie die Arbeit in Teams aus mehreren Studierenden ermöglichen, was etwa die gleichzeitige Verbindung mehrerer Benutzer zur Konsole emulierter Netzwerk-Komponenten erfordert.

4 Evaluation geeigneter Virtual Network Testbeds

Als geeignete Emulatoren haben sich im Umfeld des NetLab vor allem Cisco VIRL, GNS3 und Mininet erwiesen. Die im vorherigen Abschnitt genannten Auswahlkriterien wurden von eins für ein niedriges bis sieben für ein hohes Ranking gewichtet. Die Merkmale für die Werkzeuge wurden basierend auf Erfahrungswerten aus dem Einsatz im NetLab in Werten zwischen null (am schlechtesten) und zehn (am besten) evaluiert. Gewicht und Wert werden für jedes Kriterium multipliziert und aufsummiert, um die gewichtete Gesamtpunktzahl für jedes Werkzeug zu erhalten. Das in Abbildung 3 aufgeführte Werkzeug Cisco Modeling Labs (CML) ist lediglich der Vollständigkeit halber enthalten, da CML für den Betrieb von Cisco VIRL als Mehrplatzvariante erforderlich ist. CML und VIRL nutzen die gleiche technische Basis, jedoch ist CML wegen seines Verkaufsmodells sehr kostenintensiv und wird deshalb im NetLab nicht eingesetzt. Auch andere Emulatoren wie EVE-NG [18b] bzw. UNetLabv2 [18h] wurden noch nicht einbezogen, da sie sich zum Zeitpunkt unserer Tests noch in einer frühen Entwicklungsphase befanden. Ebenfalls wurde eNSP [18c] bislang nur testweise im NetLab eingesetzt, da es zwar eine Vielzahl der gestellten Anforderungen erfüllt, jedoch auf die Bereitstellung von Huawei-Komponenten beschränkt ist.

Zum Einsatz kommen im NetLab derzeit Cisco VIRL [18i] und GNS3, die in einer VMware vSphere 6.5 Infrastruktur betrieben werden. Eine VIRL Lizenz kostet derzeit 199 € pro Jahr und erlaubt die Virtualisierung von bis zu 20 virtuellen Cisco Geräten in emulierten Topologien. Virtuelle Geräte anderer Hersteller (z.B. Arista vEOS, Juniper JunOS, GNU/Linux) können ohne Limitierung hinzugefügt werden. Bei einer Cluster-basierten Installation können die Topologien auf mehreren Compute Nodes implementiert werden, wobei bei VIRL OpenStack für eine gleichmäßige Lastverteilung gestarteter Topologien über mehrere verfügbare VIRL Hosts sorgt. Einen wesentlichen Nachteil von VIRL stellt jedoch die geänderte Strategie von Cisco in Bezug auf den Wegfall der Academic License dar, wodurch die Kosten für die Lizenzierung erheblich steigen (vgl. [18j]). Obwohl die Cisco-eigenen Tools wie Packet Tracer, VIRL oder CML am besten geeignet sind, unsere Cisco-basierten Kurse und Laborübungen zu unterstützen, haben diese Tools derzeit große Probleme in Bezug auf den Funktionsrealismus (bei Packet Tracer), bzw. den Wartungsaufwand oder die hohen Lizenzkosten (bei VIRL und CML).

Als Alternative zu VIRL erfüllt GNS3 [18d] seit der Version 2.1.0 derzeit die meisten Anforderungen für die Erstellung virtueller Testnetzwerke im Bereich des NetLab. Zudem bietet es die größte Unterstützung für virtuelle Maschinen, Images und Network Operating Systems (NOS) unterschiedlichster Netz-Komponenten und Hersteller. Verwendbare Cisco-Images für GNS3 können aus lizenztechnischen Gründen allerdings nur von VIRL in

GNS3 importiert oder über ein Cisco CCO-Konto, z.B. durch Cisco Networking Academy-Mitglieder, in GNS3 bereitgestellt werden. Bis auf wenige Einschränkungen können mit GNS3 die in Abschnitt 3 definierten Anforderungen des NetLab umgesetzt werden. Der Ressourcen-Verbrauch ist deutlich geringer als bei VIRL, was sich in größeren realisierbaren virtuellen Netztopologien sowie einer geringeren Rüstzeit bemerkbar macht. Hierfür wurden Messungen im Vergleich zur in [Ri17] beschriebenen VIRL-Umgebung nach dem gleichen dort vorgestellten Verfahren durchgeführt. Für die Messung wurden fünf Instanzen der in [Ri17] beschriebenen Netztopologie (bestehend aus vier Arista vEOS Nodes) parallel jeweils in GNS3 und VIRL gestartet. Die Bereitstellungszeit für einen einzelnen VIRL Host lag in [Ri17] bei 315,2 Sekunden. Mit GNS3 konnte bei identischer Konfiguration auf einem einzelnen Host für die gleichen Topologien eine Bereitstellungszeit von von 173,5 Sekunden (55% der von VIRL benötigten Zeit) erzielt werden. Die aktuell in GNS3 noch fehlende Lastverteilung (Cluster-Lösung) im Vergleich zu VIRL könnte zukünftig durch die Entwicklung geeigneter Erweiterungen umgesetzt werden. Im Umfeld des NetLab sind hierzu Projekte und Abschlussarbeiten als Erweiterungen für GNS3 (Web-Interface basierend auf der GNS3 RESTful API) geplant.

Die in Abbildung 3 genannte Open-Source-Lösung Mininet [18e] findet im NetLab insbesondere für Abschlussarbeiten im Bereich SDN und NFV sowie für Master-Lehrveranstaltungen Einsatz. Durch die Verwendung von LXC Containern und OpenVSwitch als Basis für virtuelle Hosts und Netze erlaubt Mininet das Starten von großen Netztopologien auch auf Einzelplatzechnern. Eine Skalierung von Mininet über mehrere Compute Nodes (vgl. Load Balancing) wird nicht unterstützt. Die Python-API von Mininet ermöglicht zusätzlich Experimente zur Reproduzierbarkeit von aktuellen Forschungsergebnissen im Netzwerkbereich [YM17]. Dies umfasst auch die Emulation von QoS-Eigenschaften von Links (basierend auf NetEm im Linux Kernel). Für realitätsnahe Experimente steht die komplette Vielfalt von Linux-Werkzeugen auf der Command Line zur Verfügung. Auch die Anbindung an reale Netze bzw. das Internet ist möglich. Mininet unterstützt allerdings keine Bereitstellung von Images realer Netz-Hardware im Vergleich zu den anderen genannten Lösungen. Zusätzlich ist aufgrund der Realisierung von Mininet als Kommandozeilenwerkzeug und der daher fehlenden GUI die Kollaboration von Studierenden sowie die Einarbeitungszeit in Laborveranstaltungen schlechter als bei den betrachteten Alternativen GNS3 und VIRL. Aufgrund seines leichtgewichtigen Ansatzes für große Topologien und die hervorragende Eignung als SDN-Umgebung bleibt Mininet insb. für forschungsnahe Experimente im Master-Bereich trotz geringerer Bewertung im Vergleich zu GNS3 und VIRL aktuell die beste Wahl.

5 Fazit

Der Einsatz von virtuellen Netzwerk-Testbeds ermöglicht für das NetLab in verschiedenen Veranstaltungen eine gute Balance zwischen Praxis und Theorie. Durch den Einsatz von Emulatoren arbeiten und lernen Studierende mit realitätsnahen Tools, können Laborsitzungen vorbereiten, ihre Arbeit speichern und orts- sowie zeitunabhängig auch außerhalb der

Hochschule fortsetzen. Im Laufe der vergangenen Semester wurde hauptsächlich Cisco VIRL zur Realisierung von Virtual Network Testbeds im NetLab eingesetzt. Während diese Lösung unsere Anforderungen weitgehend erfüllte, führt die Umstellung des Lizenzmodells auf Einzelplatzversionen dazu, dass sich VIRL zukünftig nur schwer im NetLab verteilen und warten lässt. Ein Umstieg auf die von Cisco beworbene Mehrplatz-Alternative CML ist für uns mit zu hohen Kosten verbunden und somit nicht realisierbar. Daher wird ein Umstieg auf GNS3 evaluiert, welches seit der aktuellen Version einen Großteil unserer Anforderungen erfüllt. Zwar bietet GNS3 derzeit keine automatisierte Lastverteilung, jedoch benötigt es in Bezug auf die Skalierbarkeit im Vergleich zu VIRL weniger Ressourcen. Für skalierbare und forschungsnahe Testbeds (z.B. in Master-Lehrveranstaltungen) bleibt Mininet, insbesondere im SDN-Umfeld, das Mittel der Wahl auch wenn es insgesamt weniger unserer Anforderungen erfüllt als GNS3 und VIRL. Für die Zukunft planen wir neben GNS3 auch weitere Alternativen wie EVE-NG zu betrachten, da diese vielversprechende Ziele verfolgen. Zudem planen wir eine detaillierte Analyse der Performance und Skalierbarkeit von GNS3 im Vergleich zu Cisco VIRL durchzuführen.

Literatur

[18a] Cisco Modeling Labs, http://www.cisco.com/c/en/us/products/cloud-systems-management/modeling-labs, abgerufen am: 2018-04-16, 2018.

[18b] Emulated Virtual Environment Next Generation (EVE-NG), http://www.eve-ng.com/, abgerufen am: 2018-04-16, 2018.

[18c] eNSP - Enterprise Network Simulator, http://support.huawei.com/enterprise/en/network-management/ensp-pid-9017384, abgerufen am: 2018-04-16, 2018.

[18d] GNS3 - The software that empowers network professionals, https://www.gns3.com, abgerufen am: 2018-04-16, 2018.

[18e] Mininet - An Instant Virtual Network on your Laptop (or other PC), http://mininet.org, abgerufen am: 2018-04-16, 2018.

[18f] ns-3, https://www.nsnam.org, abgerufen am: 2018-04-16, 2018.

[18g] Packet Tracer - A free network simulation and visualization tool for the IoT era. https://www.netacad.com/about-networking-academy/packet-tracer, abgerufen am: 2018-04-16, 2018.

[18h] Unified Networking Lab v2 (UNetLabv2), http://www.routereflector.com/unetlab/, abgerufen am: 2018-04-16, 2018.

[18i] VIRL - Virtual Internet Routing Lab, http://virl.cisco.com, abgerufen am: 2018-04-16, 2018.

[18j] VIRL Academic license will no longer be available. https://learningnetwork.cisco.com/thread/103534, abgerufen am: 2018-04-16, 2018.

[Ba03] Baumgartner, F.; Braun, T.; Kurt, E.; Weyland, A.: Virtual Routers: A Tool for Networking Research and Education. SIGCOMM Comput. Commun. Rev. 33/3, S. 127–135, Juli 2003, ISSN: 0146-4833.

[Ba17] Bajpai, V.; Kühlewind, M.; Ott, J.; Schönwälder, J.; Sperotto, A.; Trammell, B.: Challenges with Reproducibility. In: Proceedings of the Reproducibility Workshop, Reproducibility@SIGCOMM 2017, Los Angeles, CA, USA, August 25, 2017. S. 1–4, 2017.

[Fl17] Flittner, M.; Bauer, R.; Rizk, A.; Geißler, S.; Zinner, T.; Zitterbart, M.: Taming the Complexity of Artifact Reproducibility. In: Proceedings of the Reproducibility Workshop, Reproducibility@SIGCOMM 2017, Los Angeles, CA, USA, August 25, 2017. S. 14–16, 2017.

[Ha12] Handigol, N.; Heller, B.; Jeyakumar, V.; Lantz, B.; McKeown, N.: Reproducible network experiments using container-based emulation. In: Proceedings of the 8th international conference on Emerging networking experiments and technologies. ACM, S. 253–264, 2012.

[KE04] Kayssi, A.; El-Haj-Mahmoud, A.: EmuNET: A Real-time Network Emulator. In: Proceedings of the 2004 ACM Symposium on Applied Computing. SAC '04, ACM, New York, NY, USA, 2004.

[MK16] Momeni, B.; Kharrazi, M.: Partov - a network simulation and emulation tool. J. Simulation 10/4, S. 237–250, 2016.

[Ob14] Obstfeld, J.; Knight, S.; Kern, E.; Wang, Q. S.; Bryan, T.; Bourque, D.: VIRL: The Virtual Internet Routing Lab. SIGCOMM Comput. Commun. Rev. 44/4, S. 577–578, Aug. 2014.

[PR16] Pizzonia, M.; Rimondini, M.: Netkit: network emulation for education. Software: Practice and Experience 46/2, S. 133–165, Feb. 2016.

[Ri17] Rieger, S.: Skalierbare virtuelle Netz-Testbedsfür Lehr- und Forschungsumgebungen am VIRL. In: 10. DFN-Forum - Kommunikationstechnologien, 30.-31 Mai 2017, Berlin, Germany. S. 125–134, 2017.

[SRP17] Seifert, C.; Rieger, S.; Pape, C.: Realization Possibilities for Virtual Networking Labs in Higher Education Courses. In: 13 th Annual International Conference on Computer Science and Education in Computer Science (CSECS). 2017.

[Ta16] Tagliacane, S. V.; Prasad, P. W. C.; Zajko, G.; Elchouemi, A.; Singh, A. K.: Network simulations and future technologies in teaching networking courses: Development of a laboratory model with Cisco Virtual Internet Routing Lab (Virl). In: 2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET). IEEE, S. 644–649, 2016.

[YM17] Yan, L.; McKeown, N.: Learning Networking by Reproducing Research Results. ACM SIGCOMM Computer Communication Review 47/2, S. 19–26, 2017.

Netztechnologien

Müller, Neumair, Reiser, Dreo Rodosek (Hrsg.): 11. DFN-Forum Kommunikationstechnologien,
Lecture Notes in Informatics (LNI), Gesellschaft für Informatik, Bonn 2018 115

Protokollgestützte Selbstbeschreibung in Zugangsnetzen

Tobias Guggemos[1], Vitalian Danciu[1], Annette Kostelezky[1]

Abstract: Die Selbstbeschreibung leitungsgebundener Anschlüsse ist für die Erstkonfiguration und Fehlersuche hilfreich. Die Modellierung eines Managementszenarios in einem Hochschulnetz erlaubt die Isolation von Fehlertypen, die mit Hilfe eines in dieser Arbeit vorgestellten Protokolls zur Selbstbeschreibung adressiert werden können. Die Untersuchung des Einsatzes dieses Protokolls in mehreren Netzen weist seine Eignung für großflächigere Nutzung auf.

Keywords: Selbstbeschreibung; LAN; VLAN; Zugangsnetz

1 Einführung

Die Zugangsnetze eines Hochschulnetzes können sich als virtuelle Netze bzw. VLANs in die Institute der Universität erstrecken, um dort Endgeräte miteinander und mit dem Internet zu verbinden. Die Nutzung des Zugangsnetzes erfolgt durch die Institute selbst, die manche Aspekte ihrer Struktur mitbestimmen und mitverwalten. Die einer virtuellen Netztopologie inhärente Flexibilität erfordert ein verteiltes Management der Zuordnung von Ressourcen wie Adressen, Portgruppen, Netzanschlüssen (Dosen) in Arbeitszimmern, VLAN-IDs auch für leitungsgebundene Anschlüsse. Zur Vermeidung von Fehlern und Erleichterung der Initialkonfiguration ist eine Selbstbeschreibung erforderlich, wie sie etwa von 802.11-Netzen bekannt ist.

Szenario
Betrachten wir ein Hochschulnetz bestehend aus einem zentral betriebenen Kernnetz, Zugangsnetzen und lokal, jeweils in den Instituten verwaltete Endgeräte (Server, Terminals, Drucker etc). Administratoren an den Instituten fordern am Service Desk des zentralen Netzbetreibers Netzressourcen an. Der Netzbetreiber schaltet die erforderlichen Anschlüsse als VLANs an den entsprechenden Switchports, weist entsprechende IP-Subnetze zu und sorgt für die Vermittlung der IP-Adressen dieser Subnetze im Hochschulnetz. Der Übergabepunkt des Dienstes sind die mit den Switchports fest verbundenen Wanddosen.

Der Administrator hat als *Besitzer* der zugewiesenen Ressourcen folgende Aufgaben bei der Inbetriebnahme: 1. Überprüfung von a) Zusammengehörigkeit der Switchports bzw. Wanddosen in einem VLAN b) Korrektheit der zugewiesenen VLAN-ID c) Korrektheit des

[1] MNM-Team, Ludwig-Maximilans-Universität München, Oettingenstr. 67, 80538 München, Germany
Email: {guggemos, danciu, kostelezky}@nm.ifi.lmu.de

zugewiesenen IP-Subnetzes und der Vermittlungsfunktion 2. Konfiguration von IP-Adressen und sonstigen Parametern für den Netzzugang an den Endgeräten 3. Anschluss von a) Servern an die Switchports b) Arbeitsplatzrechnern/Terminals und Peripherie an die Wanddosen, sowie 4. Reklamation eventueller Konfigurationsfehler an den Netzbetreiber.

Diese trivial erscheinenden Aufgaben des Besitzers stellen Herausforderungen aufgrund der Verteilung der Information über die Ressourcenzuweisung zwischen Netzbetreiber und -besitzerdomäne sowie aufgrund der Freiheiten in der dynamischen Zuweisung der Adressen und Ports: in einem Laborraum mit einer signifikanten Anzahl Wanddosen ist es selbst bei ihrer korrekten Beschriftung nicht ohne weitere Aufzeichnungen ersichtlich, an welche Dose ein Rechner mit einem bestimmten Zweck angeschlossen werden soll. Fehlerfälle können sowohl die mitgeführten Aufzeichnungen als auch die Konfiguration seitens des Netzbetreibers in Zweifel ziehen.

Beitrag und Übersicht dieser Arbeit
Die im Szenario aufgezeigten Herausforderungen werden in Abschnitt 2 in einem Modell gefasst, dass eine Unterscheidung der in dieser Arbeit betrachteten Teilprobleme erlaubt. Als Lösungsansatz wird ein Protokoll zur Selbstbeschreibung in Abschnitt 3 eingeführt und sein bisheriger praktischer Einsatz in Abschnitt 5 diskutiert. Abschnitt 6 diskutiert den Stand der Technik und verwandte Arbeiten zu Selbstbeschreibung und *Discovery*. Schließlich werden in Abschnitt 7 weiterführende Ideen für die Selbstkonfiguration in Zugangsnetzen vorgestellt.

2 Problemraum

Die Formalisierung des Szenarios in Abschnitt 1 erarbeitet die Relationen zwischen den Elementen in den verschiedenen Managementdomänen auf sowie die Informationen, die zwischen Besitzer und Betreiber (im folgenden auch „ISP“) abgeglichen werden müssen. Die besprochenen Konzepte werden in Abb. 1 illustriert.

2.1 Modell

Wir definieren die relevanten Begriffe und Rollen, beschreiben Annahmen bezüglich des Wissens der jeweiligen Rollen sowie bezüglich des Verwaltungsprozesses für Netzressourcen.

Begriffe
Die folgenden zu verwaltenden Elemente sind für den Managementanwendungsfall relevant. Großbuchstaben ’X’ verstehen sich als „Menge aller x”, ein Subskript „frei“ oder „belegt“ gibt die freie bzw. belegte Teilmenge der x an.

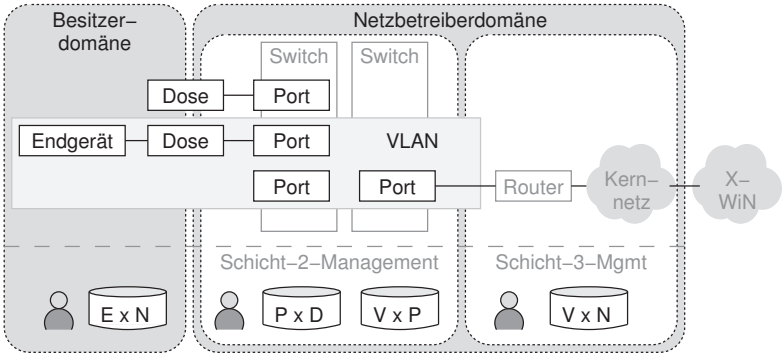


Abb. 1: Elemente und Domänen

e Endgerät: ein Rechner, der an eine Dose angeschlossen ist
d Dose: Endpunkt in der Wand, nicht am Switch
a Anschluss: die Verbindung zwischen einem Endgerät und einer Dose; kann korrekt oder falsch sein.
p Port: am Switch
v VLAN: $v \subset P$ d.h. Untermenge der Menge aller Ports
n IP-Netz: eine Menge von IP-Adressen eines IP-Netzes in einer BC-Domäne
z Zweck: eine Nutzungsabsicht für eine Ressource

Rollen
Wir unterscheiden zwischen den folgenden Rollen und ihren typischen Funktionen.

b Besitzer: Nutzer eines VLAN bzw. IP-Subnetz und Administrator der daran angeschlossenen Endgeräte
ISP/L2 Netzmanager für die Sicherungsschicht: Vergabe Portgruppe/VLAN, Konfiguration der Switches
ISP/L3 Netzmanager für die Vermittlungsschicht: Vergabe IP-Adressen, Konfiguration der Router

Zuweisungsfunktionen werden vom Besitzer ausgelöst und umfassen: 1. Zuweisung eines neuen VLAN (neue VLAN-ID) an den Besitzer. 2. Zuweisung eines Ports und Einbindung in ein bestehendes VLAN des Besitzers. 3. Umschaltung eines Ports zwischen zwei VLAN des Besitzers.

2.2 Fragestellung

Aufgabenstellung dieser Arbeit ist die Unterstützung des Besitzers bei der Erstellung korrekter Anschlüsse. Ein Anschluss a ist korrekt, wenn gilt:

Abbildung	*verantwortlich*	*Fehlerbeispiel*
$E \mapsto N$	b	falsche IP-Adresse, falsche Netzmaske
$N \mapsto V$	ISP/L3	Port vergessen, Port in falschem VLAN
$V \mapsto P$	ISP/L2	falscher Eintrag in RT
$P \mapsto D$	ISP/b	falscher Patch, falsche Beschriftung

Tab. 1: Abbildungen und Beispiele für Fehler

- e hat eine IP-Adresse aus n
- n wird in v geroutet
- p gehört zu v
- p führt zu d

Die Abbildungen zwischen Netzelementen sind in Tab. 1 dargestellt.

Einschränkungen
Wir schließen die Betrachtung der folgenden Aspekte aus: 1. Eigenschaften von Ports (z.B. tagged/untagged): ein erfolgreich zugewiesener Port ist eingeschaltet und zweckmäßig konfiguriert; sonst gilt er als nicht erfolgreich zugewiesen. Wir nehmen ferner an, dass der ISP keine von Besitzern nutzbare Informations- und Konfigurationsdienste bereitstellt. Dazu gehören 2. Dienste seitens des ISP/L2 (z.B. Identifizierung des Switch mit LLDP) sowie 3. Dienste seitens des ISP/L3 (z.B. DHCP). 4. Die Rückgabe von Ressourcen an den ISP wird nicht berücksichtigt: diese Arbeit fokussiert sich auf die korrekte Zuweisung und Konfiguration.

Systematik der Fragestellung
Die in Tab. 2 dargestellten Fälle repräsentieren das Vorkommen von Fehlern in den in Tab. 1 aufgezählten Abbildungen. Ein Mechanismus zur Selbstbeschreibung sollte in der Lage sein, einen Teil dieser Fälle direkt zu unterstützen (z.B. Fall 3) und manche mittelbar (z.B. kann für Fall 9 eine Überprüfung der Schicht-3-Konnektivität erfolgen). Ambiguität liegt vor, wenn aufgrund der Kenntnis der Broadcast-Domäne, ihrer VLAN-ID sowie des Netzpräfix eine scheinbar korrekte Funktionsweise denkbar ist.

3 Protokoll zur Selbstbeschreibung

Der Name des Protokolls, *A-NetBeacon*, deutet auf sein Funktionsprinzip hin, durch periodische Nachrichten in einer Broadcast-Domäne der Sicherungsschicht über die Eigenschaften dieser Broadcastdomäne zu informieren. Das wichtigste Ziel beim Entwurf des Protokolls ist ein möglichst großer Bereich seiner Anwendbarkeit. Es macht daher keine Annahmen bezüglich der Vermittlungsschicht, sondern wird direkt in Rahmen der Sicherungsschicht transportiert. Die Annahmen bezüglich der Sicherungsschicht beschränken sich auf die durch die Ethernet-Hardware gegebenen Eigenschaften. Weiterhin werden die übertragenen Daten auch in einer menschenlesbaren Fassung in den Nachrichten kodiert, um eine

	N × V	V × P	P × D	N × E	Rolle	Ambig?	Fehlerbeispiel
1	.	.	.	.	–	-	keine Fehler
2	.	.	.	✘	b	-	e falsch konfiguriert
3	.	.	✘	.	b	-	Dose falsch beschriftet
4	.	.	✘	✘	b	✔	2 und 3
5	.	✘	.	.	L2	-	Port in falschem VLAN
6	.	✘	.	✘	b, L2	✔	2 und 5
7	.	✘	✘	.	L2	✔	falsches VLAN an falsch beschrifteten Port
8	.	✘	✘	✘	b, L2	✔	4 und 5: Schicht-2-Fehler
9	✘	.	.	.	L3		falscher Eintrag in RT für n in v
10	✘	.	.	✘	b, L3	✔	e falsch konfiguriert im falschen Netz
11	✘	.	✘	.	L2, L3	✔	falsches Netz an falscher Dose
12	✘	.	✘	✘	alle	✔	2 und 10
13	✘	✘	.	.	L2, L3	✔	falsches Netz im falschen VLAN
14	✘	✘	.	✘	alle	✔	2 und 13
15	✘	✘	✘	.	L2, L3	✔	Konfiguration in ISP-Domäne falsch
16	✘	✘	✘	✘	alle	✔	alle Fehler

Tab. 2: Betrachtete Fehlerfälle: ✘ bezeichnet Fehler

leichte Lesbarkeit der Selbstbeschreibungsnachrichten mit generischen Werkzeugen zu ermöglichen.

Funktion
Eine “Beacon”-Nachricht wird im Ethernet-Broadcast über einen Anschluss des Besitzers im VLAN verbreitet (vgl. Abb. 2). Sie kann an allen anderen Dosen bzw. Ports empfangen werden und lässt die Gruppenzugehörigkeit von Ports nachweisen. Die Information in den Nachrichten gibt direkte Hinweise auf mindestens die folgenden Fehlerfälle aus Abschnitt 2, Tab. 2: die Fälle (2, 3, 4) in alleiniger Verantwortung des Besitzers, und isolierte Fehler (5, 7) in der Schicht-2-Konfiguration des Netzbetreibers. Grundlage für weitere Diagnoseschritte wird für die Fälle (9, 10, 13) gegeben.

Sicherheit
Zur Wahrung von Integrität und Authentizität der für das Netzmanagement genutzten Informationen ist ein Challenge-Response-Verfahren vorgesehen mit dem Clients signierte Nachrichten vom Server anfordern können. Der Client sendet eine Nonce in der Anfrage, die von dem Server zusammen mit einem Zeitstempel und der Selbstbeschreibungsinformation signiert wird. Das Verfahren bietet Schutz vor Replay-Angriffen, bei denen die Nachrichten eines Servers in ein anderes Netz kopiert werden.

Datenmodell
Die gesendeten Daten sind Name-Wert-Paare, die im Prototyp als LLDP-Tripel (Typ, Länge, Wert) codiert werden. Dieses erweiterbare Schema umfasst derzeit als optionale Felder 1. VLAN-ID 2. VLAN Name 3. Benutzerdefinierter Freitext 4. IPv4 Netz 5. IPv6 Netz 6. E-Mail-Adresse des Domänenbesitzers 7. Authentifizierung (siehe Abschnitt 3) 8. Repräsentation aller Inhalte in ASCII.

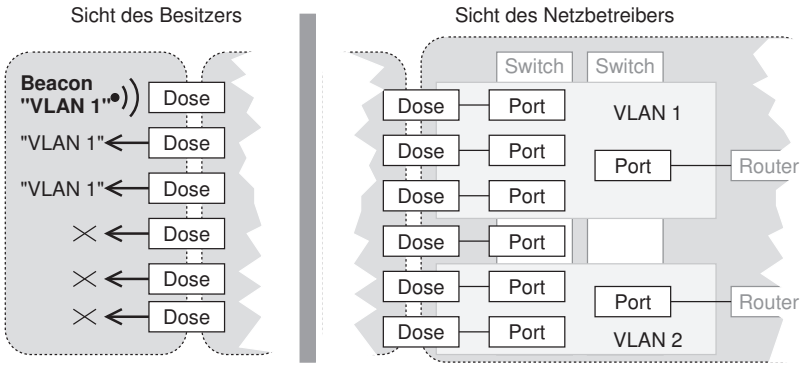


Abb. 2: Funktionsprinzip und Sichten

Um die Unterdrückung des Broadcast durch LLDP-fähige Switches zu vermeiden benutzt *A-NetBeacon* einen *Local Experimental Ethertype* mit Typcode `0x88b5`.

Prototyp
Ein Prototyp wurde in C implementiert[2] und auf einem Kleinstrechner (Banana Pi mit einem angeschlossenen Bildschirm erprobt. Der Prototyp implementiert sowohl den Server als auch eine Client-Anwendung, welche die in LLDP transportierten Informationen einfach menschenlesbar auf dem Bildschirm darstellt. Neben der Verwendung des Clients ist es aber auch möglich, die Informationen mittels eines Werkzeugs zur Netzanalyse (z.B. *tcpdump*) anzuzeigen. Abb. 3 zeigt beide Fälle.

4 Nutzung des *A-NetBeacon*

Die Auslöser zur Nutzung des *A-NetBeacon* sind Abweichungen des Erwartungswertes von ExN (durch Störungsmeldungen) oder geplante Änderungen in VxN oder PxD

Daraufhin entwickelt der Administrator der Besitzerdomäne eine Diagnosestrategie (für Störungen) bzw. eine Überprüfungsstrategie (für Änderungen). Dazu gehören die zu überprüfenden VLANs (inkl. IP-Adressbereichen) und die dazu gehörenden Dosen zur Prüfung von VxP. PxD kann als korrekt angenommen werden, wenn es dem Erwartungswert von ExN entsprechend konfiguriert sein sollte.

Anschließend wählt der Administrator der Besitzerdomäne eine oder mehrere Platzierungen für den Sender des *A-NetBeacon* mit entsprechenden Parametern, so dass die Überprüfung bzw. die Diagnose unterstützt wird.

Aus der Diagnose-/Überprüfungsstrategie ergeben sich die zu erwartenden empfangbare *A-NetBeacon* an Endgerät E und lässt wiederum einen entsprechenden Ergebnisschluss zu.

[2] Die Implementierung steht quelloffen zur Verfügung: `https://github.com/mnm-team/LANbeacon`

```
08:07:48.380203 b8:27:eb:0d:b2:0e (oui Unknown) > Broadcast, ethertype Unknown (0x88b5),
 length 255:
        0x0000:  0207 04b8 27eb 0db2 0e04 0703 b827 eb0d  ....'........'..
        0x0010:  b20e 0602 0014 fe18 cc4d 55cb 0a99 3300  .........MU...3.
        0x0020:  180a 9907 8019 81bb d600 1881 bbe4 0018  ................
        0x0030:  fe12 cc4d 55cd 7262 6740 6966 692e 6c6d  ...MU.rbg@ifi.lm
        0x0040:  752e 6465 fe06 cc4d 55c8 03ca fe0c cc4d  u.de...MU......M
        0x0050:  55c9 4946 4920 4e65 747a fe93 cc4d 55d9  U.IFI.Netz...MU.
        0x0060:  4950 7634 3a20 4946 4920 4e65 747a 3a20  IPv4:.IFI.Netz:.
        0x0070:  3130 2e31 3533 2e35 312e 302f 3234 2c20  10.153.51.0/24,.
        0x0080:  3130 2e31 3533 2e37 2e31 3238 2f32 352c  10.153.7.128/25,
        0x0090:  2031 3239 2e31 3837 2e32 3134 2e30 2f32  .129.187.214.0/2
        0x00a0:  342c 2031 3239 2e31 3837 2e32 3238 2e30  4,.129.187.228.0
        0x00b0:  2f32 342e 2045 6d61 696c 3a20 7262 6740  /24..Email:.rbg@
        0x00c0:  6966 692e 6c6d 752e 6465 2e20 564c 414e  ifi.lmu.de..VLAN
        0x00d0:  2d49 443a 2039 3730 2e20 564c 414e 2d4e  -ID:.970..VLAN-N
        0x00e0:  616d 653a 2049 4649 204e 6574 7a2e 2000  ame:.IFI.Netz...
        0x00f0:  00                                       .
```

(a) Informationsanzeige mittels tcpdump

(b) Prototyp auf Kleinstrechner im Einsatz

Abb. 3: *A-NetBeacon* im Einsatz

5 Evaluation in der Praxis

Zur Bewertung der Einsatzfähigkeit wird *A-NetBeacon* in Produktivnetzen am Institut für Informatik der LMU erprobt. Abb. 4 illustriert diesen Einsatz in realen Besitzerdomänen sowie in virtualisierter Laborinfrastruktur.

Einsatz 1: Eine Besitzerdomäne.
Im Fall (siehe (1) in Abb. 4), der die Konzeption dieser Arbeit ausgelöst hat, soll ein Administrator in Besitzerrolle die korrekte Konfiguration durch den ISP testen und überprüfen. Die *A-NetBeacon*-Serveranwendung wird auf einem unter der Kontrolle des Administrators stehenden Server mit einer Schnittstelle in allen VLANs der Domäne betrieben und anschließend an allen im VLAN zugänglichen Dosen empfangen werden.

Anekdote aus dem Alltag eines Sysadmins:
Im Regelfall sind alle genutzten TP-Zimmerdosen mit der dahinterliegenden VLAN-Konfiguration in für den Mitarbeiter „übersetzter Form“ beschriftet, z.B *Laptop*. Für die Bereitstellung eines weiteren Anschlusses für einen Laptop an einer unbeschrifteten Dose wird zur Prüfung der aktuellen Konfiguration der tragbare Kleinstrechner mit dem *A-NetBeacon*-Client angeschlossen und festgestellt, dass die Dose zwar gepatched aber in ein fremdes VLAN eingebunden wurde. Dadurch konnte eine explizite Serviceanfrage an den ISP gestellt werden, wohingegen ohne das *A-NetBeacon* im Falle einer korrekten Konfiguration unnötige Arbeit verursacht worden wäre.

Einsatz 2: Überlappende Besitzerdomänen.
Neben der Verwendung in der „eigenen“ Domäne ist in manchen Szenarien auch eine Kooperation mehrerer Besitzer nötig, beispielsweise wenn Netze von Besitzern verschiedener Domänen geteilt werden. Dieser Anwendungsfall wurde am Beispiel des Institutsnetzes der Informatik der LMU München erprobt (siehe (2) in Abb. 4). Dabei wurde die Serveranwendung des *A-NetBeacon* auf einem am sogenannten „IFI Multinet“ angeschlossenen Geräten gestartet und konnte dann domänenübergreifend empfangen werden.

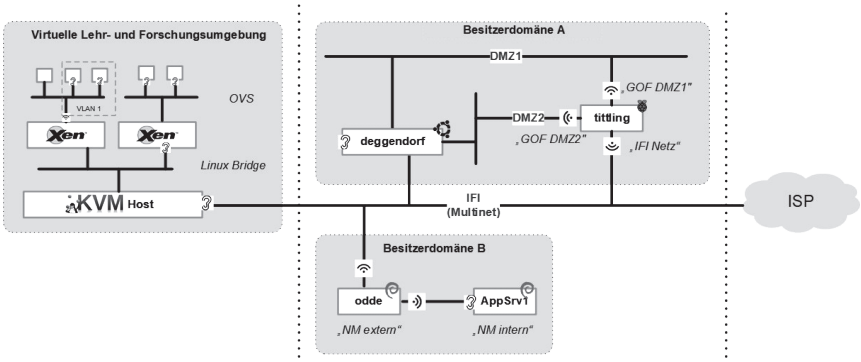


Abb. 4: *A-NetBeacon* in unterschiedlichen Besitzerdomänen

Einsatz 3: In der Rolle eines Netzbetreibers.
Zur Untersuchung der Eignung für den ISP-seitigen Einsatz agiert der Besitzer der Domäne A im dritten Szenario (siehe (3) in Abb. 4) als ISP für ein in sich geschlossenes Lehr- und Forschungsnetz und sendet *A-NetBeacon*-Nachrichten, die von allen an der virtuellen Infrastruktur angeschlossenen Geräten empfangen werden.

Einsatz 4: Virtuelle Netzkomponenten.
Aufbauend auf Szenario 3 wurde das Beacon auch in einem virtuellen Netz innerhalb einer geschichteten Labor- und Lehrinfrastruktur [DGK] getestet, um Einschränkungen an virtuellen Netzkomponenten ausschließen zu können. Dabei konnten *A-NetBeacon*-Nachrichten über die virtuellen Switches (als Linux Bridges und OpenVSwitch, konfiguriert mit tagged, untagged und ohne VLAN-ID) verteilt und empfangen werden. Lediglich beim Zusammenspiel von getaggten und ungetaggten VLANs innerhalb einer Infrastruktur kam zu Duplizierung der Rahmen auf Grund der Broadcast-Implementierung von OpenVSwitch.

Die Nutzung des Protokolls hat in den bisherigen Versuchen Besitzer und Netzbetreiber entlastet, indem die Prüfung korrekter Anschlüsse Serviceanfragen vermeidet. Für Laborinfrastruktur kann der Besitzer in ISP-Rolle gegenüber z.B. Praktikumsbetreuern mittels des Protokolls die Netze und Konfigurationsoptionen bekannt geben und Rückfragen vermeiden. Ein Einsatz in größerem Rahmen wird derzeit geprüft.

6 Themenverwandte Arbeiten

Das durch *A-NetBeacon* angesprochene Problem kann in den Bereich der Topologieerkennung bzw. *Topology Discovery* eingeordnet werden. Hassan Gobjuka [Go10] stellt einen Algorithmus zur Ermittlung der Netztopologie heterogener Netze vor. Er beschreibt die Ermittlung in VLAN als deutlich komplexer als in physischen Netzen (LAN) und bezeichnet sie als *NP-hart*. Andere Arbeiten (exemplarisch: [BDF04; Br00]) beschreiben Topologieerkennung in lokalen Netzen und Ermittlung der genutzten IP-Netze im Rahmen

von *Network Discovery*. Schichtübergreifende Erkennung ist durch datenbankgestützte aktive Verfahren möglich [LOG01] z.B. gestützt auf Inhalte der *Management Information Base* (MIB [IE09]) von SNMP-Agenten, aber auch Ansätze, die den Spannbaum [St02] bzw. die *Forwarding Tables* [SWS05] zur Topologieermittlung.

Diese Verfahren untersuchen die Gesamttopologie des Netzes, erfordern Zeit und stehen bei der Konfigurationsaktivität, wie sie im Szenario besprochen wird, nicht unmittelbar zur Verfügung. Häufig eingesetzte Techniken zur Überprüfung eines korrekten Anschlusses nutzen „Bordmittel“ der Netzdiagnose (*ping, traceroute, portscan*, etc.). Das an einer Dose angeschlossene VLAN ist bei *tagged* Rahmen anhand der VLAN-ID im 802.1Q Header möglich, sollte der Verkehr im Zugangsnetz in solchen Rahmen transportiert werden.

Das in dieser Arbeit vorgestellte *A-NetBeacon*-Protokoll schließt diese Lücke, indem es ein Informationssystem für die Besitzerdomäne bereitstellt. Bereits vorhandene Funktionen der Netzkomponenten, vor allem *Link Layer Discovery Protocol* (LLDP) 802.1 AB [IE15] und seine herstellerspezifischen Varianten (Cisco CDP, Microsoft LLTP) könnten trotz ihres Fokus auf automatische Erkennung zwischen Geräten eingesetzt werden, erfordern jedoch administrativen Zugriff auf alle betroffenen Netzkomponenten.

7 Zusammenfassung und Ausblick

Die Erstellung eines korrekten Anschlusses eines Endgerätes im Zugangsnetz kann durch Selbstbeschreibung mit dem vorgestellten *A-NetBeacon*-Protokolls auch ohne die Erfordernis administrativer Rechte im Zugangsnetz unterstützt werden. Seine Erprobung in der Praxis weist es als Hilfestellung des realen Betriebs aus.

Unser Problemmodell weist aber auch Fälle auf, die nur mittelbar durch Einsatz des Protokolls analysiert werden können und somit Kandidaten für weitere Untersuchungen darstellen. Darüber hinaus sind Weiterentwicklungen basierend auf der Selbsbstbeschreibungsfunktion denkbar, etwa der Abgleich der Konfiguration von Managementdiensten (z.B. DHCP, Intrusion Detection, Paketfilter). Die Integration des Protokolls in die Software der Netzkomponenten würde die Überprüfung der Selbstbeschreibungsdaten erlauben. Solche Entwicklungen könnten die Haltung von Managementdaten konsolidieren und möglicherweise Ansätze für die zusammengesetzten Fälle des Problemmodells erlauben.

Danksagung
Die Autoren bedanken sich herzlich bei Herrn BSc Bitzer, der leider an der Erstellung dieses Papiers nicht teilnehmen konnte. Im Rahmen seiner Bachelorarbeit [Bi17] arbeitete er an der Konzeption des Protokolls mit und entwickelte den Prototypen, der bei unseren Versuchen eingesetzt wurde.

Literatur

[BDF04] Black, R.; Donnelly, A.; Fournet, C.: Ethernet topology discovery without network assistance. In: Proceedings of the 12th IEEE International Conference on Network Protocols, 2004. ICNP 2004. S. 328–339, Okt. 2004.

[Bi17] Bitzer, D.: LAN-Beacon: Ein Protokoll zur authentifizierten Selbstbeschreibung lokaler Netze, Ludwig-Maximians-Universität München, Juni 2017.

[Br00] Breitbart, Y.; Garofalakis, M.; Martin, C.; Rastogi, R.; Seshadri, S.; Silberschatz, A.: Topology discovery in heterogeneous IP networks. In: Proceedings IEEE INFOCOM 2000. Conference on Computer Communications. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (Cat. No.00CH37064). Bd. 1, 265–274 vol.1, 2000.

[DGK] Danciu, V.; Guggemos, T.; Kranzlmüller, D.: Schichtung virtueller Maschinen zu Labor- und Lehrinfrastruktur. In: 9. DFN Forum Kommunikationstechnologien. Bd. 2016. GI–Edition Lecture Notes in Informatics, Rostock Deutschland.

[Go10] Gobjuka, H.: Topology Discovery for Virtual Local Area Networks. In: 2010 Proceedings IEEE INFOCOM. S. 1–5, März 2010.

[IE09] IEEE: IEEE Standard for Local and metropolitan area networks- Virtual Bridged Local Area Networks Amendment 8: Management Information Base (MIB) Definitions for VLAN Bridges. IEEE Std 802.1ap-2008 (Amendment to IEEE Std 802.1Q-2005)/, S. c1–323, März 2009.

[IE15] IEEE: IEEE Standard for Local and metropolitan area networks– Station and Media Access Control Connectivity Discovery Corrigendum 2: Technical and Editorial Corrections. IEEE Std 802.1AB-2009/Cor 2-2015 (Corrigendum to IEEE Std 802.1AB-2009)/, S. 1–68, März 2015.

[LOG01] Lowekamp, B.; O'Hallaron, D.; Gross, T.: Topology Discovery for Large Ethernet Networks. In: Proceedings of the 2001 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications. SIGCOMM '01, ACM, San Diego, California, USA, S. 237–248, 2001, ISBN: 1-58113-411-8.

[St02] Stott, D. T.: Layer-2 path discovery using spanning tree MIBs. Avaya Labs Research, Avaya Inc 233/, 2002.

[SWS05] Sun, Y.; Wu, Z.; Shi, Z.: The physical topology discovery for switched Ethernet based on connections reasoning technique. In: IEEE International Symposium on Communications and Information Technology, 2005. ISCIT 2005. Bd. 1, S. 44–47, Okt. 2005.

Müller, Neumair, Reiser, Dreo Rodosek (Hrsg.): 11. DFN-Forum Kommunikationstechnologien,
Lecture Notes in Informatics (LNI), Gesellschaft für Informatik, Bonn 2018 125

Netzstrukturen für Weitverkehrsnetze

Andreas Hanemann[1]

Abstract: Sobald eine neue Generation eines Weitverkehrsnetzes entworfen wird, sind grundlegende Entscheidungen über den Netzaufbau zu treffen. Dabei sind Kriterien wie die Bereitstellung von Dienstangeboten entsprechend dem Anwenderbedarf, hohe Ausfallsicherheit, kurze Verzögerungen und geringe Kosten zu beachten. In diesem Beitrag werden daher die verschiedenen Möglichkeiten für den Aufbau der Netzknoten und die Rollen der unterschiedlichen Netzebenen betrachtet.

Keywords: Weitverkehrsnetz; Ausfallsicherheit

1 Einleitung

Weitverkehrsnetze stellen eine wichtige Infrastruktur dar, an die hohe Anforderungen hinsichtlich der Betriebsstabilität und den Eigenschaften der Dienstangebote gestellt werden. Manche Ziele wie eine hohe Ausfallsicherheit bei technischen Störungen oder geringe Anschaffungs- und Betriebskosten sind dabei schon lange bekannt. In den letzten Jahren deutlich relevanter geworden ist die Sicherstellung der Betriebsstabilität bei Distributed Denial of Service (DDoS)-Angriffen. Diese Problematik hat sich insofern verschärft, dass Angreifer mit Methoden wie Amplification-Angriffen in der Lage sind, sehr hohe Datenraten von mehreren hundert Gigabit/s zu erzeugen (siehe [Ak]). Diese Angriffe betreffen dann deutlich mehr als nur das primäre Angriffsziel.

Um die genannten Ziele zu erreichen, müssen beim Aufbau einer neuen Netzgeneration grundsätzliche Entscheidungen getroffen werden. Es muss festgelegt werden, wie die Standorte (Points of Presence, PoPs) intern aufgebaut werden und ob es unterschiedliche Standorttypen gibt, die sich im Aufbau und in der Leistungsfähigkeit unterscheiden. Für die Ausfallsicherheit spielen Redundanzkonzepte eine wichtige Rolle, wobei man an dieser Stelle Schutzmechanismen auf unterschiedlichen Ebenen betrachten muss.

Der vorliegende Beitrag zielt darauf ab, die Vor- und Nachteile der Möglichkeiten zu beleuchten und somit hilfreich für eine Entscheidungsfindung unter Beachtung der Randbedingungen eines gegebenen Netzes und seiner Kunden zu sein. Die Überlegungen treffen dabei auf Forschungsnetze zu, aber gelten ebenso für kommerzielle Netze. Durch die Konvergenz der Netze mit dem Internet Protocol bzw. Ethernet-Schnittstellen als Basis

[1] FH Lübeck, Mönkhofer Weg 239, 23562 Lübeck, andreas.hanemann@fh-luebeck.de

aller Dienste sind Unterschiede zwischen diesen Netztypen nicht mehr so groß wie in der Vergangenheit.

In diesem Papier wird davon ausgegangen, dass man klar zwischen verschiedenen Netzgenerationen unterscheidet. Das heißt, man schreibt das Netzwerk nach einigen Jahren komplett neu aus und vergibt dann die Ausstattung der optischen Technik an einen Anbieter, die Router an einen weiteren Anbieter und nimmt ggf. noch Switches von einem dritten Anbieter hinzu. Andere Vorgehensweisen wie die Beauftragung von zwei DWDM-Techniklieferanten sind (zumindest im Wissenschaftsumfeld) unüblich und führen an den Übergabepunkten zu zusätzlichen Schwierigkeiten bei der Konfiguration.

2 Angebotene Netzzugänge

Bei der Betrachtung der Weitverkehrsnetze kann man die Anschlussmöglichkeiten für die Kunden als Ausgangspunkt nehmen. Dabei sind mit Kunden nicht Einzelpersonen gemeint, sondern angeschlossene Einrichtungen.

OPN: Mit Optical Private Network wird ein Szenario bezeichnet, bei dem ein Kunde einen derart hohen Bitratenbedarf hat, dass für ihn eine dedizierte Wellenlänge geschaltet wird, z.B. mit 10 Gbit/s. Dieses ist insbesondere dann sinnvoll, wenn über Monate und Jahre hinweg sehr viel Datenverkehr zwischen festen Partnereinrichtungen ausgetauscht wird. In diesem Fall erhält der Kunde einen direkten Anschluss an den *ROADM* (Reconfigurable Optical Add/Drop Multiplexer), d.h. die optische Technik, des PoP. Im Wissenschaftsumfeld sind dedizierte Wellenlängen für die Auswertung der LHC-Versuche am CERN ein bekanntes Beispiel.

Switching: Bei Kunden mit einem mittleren Bitratenbedarf von z.B. 2 Gbit/s wäre es ungünstig, exklusiv eine Wellenlänge mit der Kapazität von 10 Gbit/s zu reservieren, weil einige Kapazität dann nicht genutzt würde. Stattdessen ist es sinnvoll zu versuchen, die Datenverkehre von mehreren Kunden dieser Art zu bündeln, um eine Wellenlänge so gut wie möglich auszulasten. Als Nebeneffekt kann man folglich den Kunden auch attraktivere Preise anbieten, als wenn man ihnen eine exklusive Wellenlänge in Rechnung stellen müsste. Auch wenn eine solche Aggregation hier als Switching bezeichnet wird (alternativ wird auch von *Grooming* gesprochen), muss dieses nicht unbedingt über separate Geräte (Switches) realisiert werden. Oftmals bietet auch der ROADM über seine Interface Cards schon entsprechende Möglichkeiten.

Routing: Kunden mit kleinerem Bitratenbedarf werden üblicherweise an Router am PoP angebunden. Durch die Anbindung an Router sind auch Layer-3-VPNs möglich, die mit MPLS (Multiprotocol Label Switching) realisiert werden.

3 Möglichkeiten für die Points of Presence

Eine zentrale Frage bei der Konzeption von Weitverkehrsnetzen ist, wie man die PoPs aufbaut. Hierbei gibt es konkurrierende Konzepte, bei denen man sich üblicherweise für eines entscheiden muss.

3.1 All Optical PoPs

Das Konzept der meisten DWDM-Technikhersteller soll als *All Optical PoP* bezeichnet werden (siehe Abb. 1). Die Grundidee ist es dabei, die Umwandlungen von optischen in elektrische Signale und zurück (*optisch-elektrisch-optisch, OEO*) möglichst zu vermeiden.

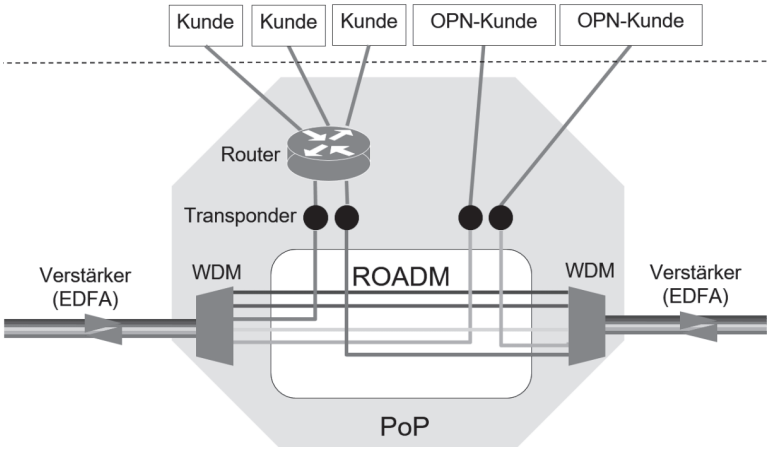


Abb. 1: Aufbau eines PoP nach All-Optical-Prinzip

Nehmen wir an, es gebe einen Standort, der mit zwei anderen Standorten verbunden ist[2]. Dieses ist in der Zeichnung mit den Verbindungen nach links und rechts angedeutet, wobei die Regenbogenfarben die Verwendung unterschiedlicher Wellenlängen darstellen sollen. Während die Wellenlängen nach links und nach rechts über je eine Glasfaser übertragen werden, ermöglichen die DWDM-Systeme es innerhalb des PoP auf einzelne Wellenlängen zuzugreifen, was als Auffächerung der Regenbogenfarben dargestellt ist.

DWDM-Systeme im Praxiseinsatz können beispielsweise wie die vom DFN verwendete Technik des Anbieters ECI Telecom 88 verschiedene Wellenlängen auf einer Glasfaser zu anderen PoPs übertragen, wobei gängige Bitraten pro Wellenlänge aktuell 10 oder 100 Gbit/s sind. Um die anderen PoPs zu erreichen, ist es notwendig alle 80 km optische Verstärker aufzubauen, wobei EDFAs (Erbium Doped Fiber Amplifier) der übliche Verstärkertyp sind.

[2] Auf Standorte mit mehr als zwei Glasfaserverbindungen zu anderen Standorten und mit unterschiedlichen Fähigkeiten von ROADMs zum Schalten von Verbindungen in diese verschiedenen Richtungen wird in diesem Papier nicht eingegangen, siehe dazu z.B. [AGN12].

Bei den Wellenlängen muss man nun unterscheiden, dass es Wellenlängen gibt, die Daten für den betrachteten PoP übertragen, und andere, die für Transitdatenverkehr genutzt werden. Der Transitdatenverkehr hat also nichts mit dem PoP zu tun und man möchte ihn daher auf der optischen Ebene belassen. Das heißt, die Signale können zwar auf optischer Ebene verstärkt werden, aber sie sollen nicht mit Hilfe von Transpondern auf die elektrische Ebene gewandelt werden. Hiermit möchte man die Kosten für die Transponder einsparen.

In grau dargestellt sieht man die Verwendung einer Standardwellenlänge. Diese kommt vor, wenn man Datenverkehr zum lokalen Router hat. Dieser muss mit Hilfe von Transpondern von einer farbigen (d.h. nicht-Standard) Wellenlänge in die Standardwellenlänge gewandelt werden. Genauso ist eine Umwandlung in die Standardwellenlänge erforderlich, wenn man dedizierte Wellenlängen für Kunden mit hohem Bitratenbedarf anbietet. Diese Kunden werden hier als OPN-Kunden bezeichnet.

3.2 All Optical PoPs mit farbigen Routerinterfaces

Eine Abwandlung des All Optical PoPs stellt die Variante mit farbigen Routerinterfaces dar, die in Abb. 2 dargestellt ist. Hierbei ist ein Detail anders, nämlich dass die Interfaces des Routers hinsichtlich der Wellenlänge konfigurierbar und damit keine Transponder notwendig sind. Der Router sendet also gleich auf farbigen Wellenlängen.

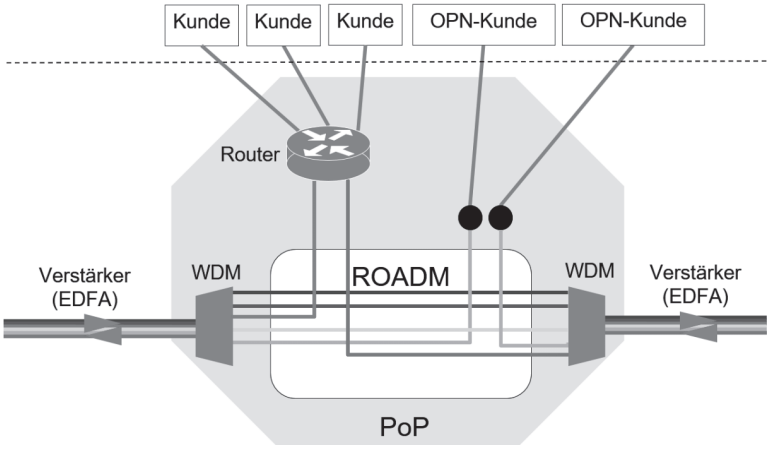


Abb. 2: Aufbau eines PoP nach All-Optical-Prinzip mit farbigen Routerinterfaces

Dieser Unterschied hat jedoch weitreichende betriebliche Auswirkungen, da nun vom Router aus optische Signale direkt zum DWDM-System geschickt werden. Hierbei müssen dann entweder die optischen Systeme unterschiedlicher Hersteller zusammenarbeiten, was noch nicht als etablierte Lösung anzusehen ist (siehe Abschnitt 5.1 im Ausblick) oder man muss sowohl die Router- als auch die DWDM-Technik von einem Anbieter beziehen. Bei der zweiten Möglichkeit schränkt man die Auswahlmöglichkeiten zwischen

den möglichen Anbietern stark ein (vgl. Cisco IPoDWDM-Konzept [Ci]), wobei diese mit besseren Möglichkeiten für ein integriertes Management der Netzebenen werben.

3.3 Digital ROADMs

Das Konzept des *Digital ROADMs* unterscheidet sich deutlich vom All Optical-Konzept. Der Digital ROADM wandelt, wie in der Abb. 3 dargestellt, sämtliche Wellenlängen in die elektrische Ebene um. Dieses wird mit einem sog. *Photonic Integrated Circuit (PIC)* erreicht, der optische Bauteile, die sonst einzeln verwendet werden, in einem speziellen Chip auf kleiner Fläche enthält. Die ersten 2005 erhältlichen PICs integrierten dabei beispielsweise die Komponenten für 10 Wellenlängen, wobei jede Wellenlänge 10 Gbit/s übertragen kann. Nachdem die Datenrate pro PIC anfangs also 100 Gbit/s war, sind inzwischen 2,4 Tbit/s möglich [Gi]. Durch dieses Konzept muss nicht mehr zwischen lokal terminierenden Wellenlängen und solchen für den Transitverkehr unterschieden werden, was die Konfiguration erleichtert.

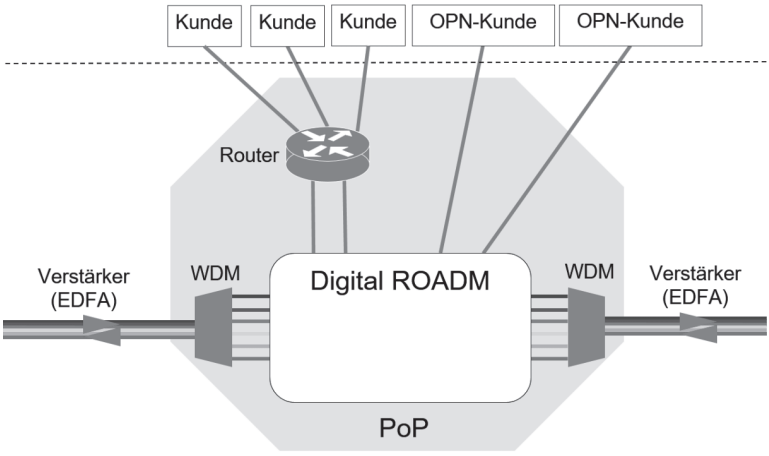


Abb. 3: Aufbau eines PoP mit Digital ROADM

Bei der Abwägung, welches Konzept für das eigene Netz besser geeignet ist, muss man sich ansehen, was man für Standorte hat. Wenn der übliche Standort viel lokal terminierenden Datenverkehr aufweist, dann ist dieses günstiger für das Digital ROADM-Konzept. An dieser Stelle hat man nur geringe Zusatzkosten für eigentlich nicht benötigte OEO-Umwandlungen, die indirekt auch zu den Kosten des PICs beitragen. Dagegen ist das All Optical-Konzept dann vorteilhaft, wenn man viele Standorte mit wenig lokal terminierendem Verkehr hat. In diesem Szenario hat man nur relativ geringe Kosten für die Transponder. Wenn man sich die Strukturen und Standorte verschiedener Netze ansieht, dann ist aus dieser Überlegung heraus verständlich, warum sich GEANT und Internet2/ESnet für ein Digital ROADM-Konzept entschieden haben, viele nationale Netze (neben DFN z.B. SWITCH,

GARR, SURFnet, PIONIER) dagegen für All Optical-Lösungen. In kommerziellen Netzen sind ebenfalls beide Konzepte im Einsatz, z.B. bei einem bekannten Provider ein Digital ROADM-Konzept für die Verbindung zwischen den europäischen Ländern, aber ein Netz gemäß All Optical-Konzept innerhalb von dessen Heimatland.

Es sei ergänzend noch erwähnt, dass auch eine Mischform möglich ist. Wenn man beispielsweise dreißig Wellenlängen verwendet und fünf von den Wellenlängen sollen an einem Standort lokal terminieren, dann kann man für diese fünf und weitere fünf eine OEO-Umwandlung mit dem PIC gemäß Digital ROADM-Konzept vorsehen. Für die anderen zwanzig Wellenlängen kann man eine optische Durchschaltung (*Optical Bypass*) konfigurieren, so dass man für diese keine PICs benötigt und entsprechende Kosten einspart. An dieser Stelle verliert man jedoch auch ein Stück an Flexibilität, wenn zukünftig Wellenlängen von diesen lokal terminieren sollen.

3.4 Routerless PoPs

Bei den bisherigen Szenarien war es so, dass in jedem Fall neben der DWDM-Technik ein Router vorhanden war. Es besteht jedoch auch die Möglichkeit, Standorte ohne Router zu betreiben, sog. *Routerless PoPs* (siehe Abb. 4). Das bedeutet, dass der Datenverkehr von Kunden an diesen PoPs entweder aus dedizierten Wellenlängen besteht oder mit Hilfe einer Aggregationsplattform zusammengeführt und an einen anderen Standort weitergeleitet wird. Dieses ist in der Abbildung als separater Switch dargestellt, aber ist üblicherweise auch als Teil der DWDM-Plattform realisierbar. Damit wird der Datenverkehr von solchen Kunden in jedem Fall zunächst zu einem vordefinierten Standort mit Router weitergeleitet. Das führt ggf. zu höheren Latenzen, abhängig davon, welches Ziel die einzelnen Pakete haben.

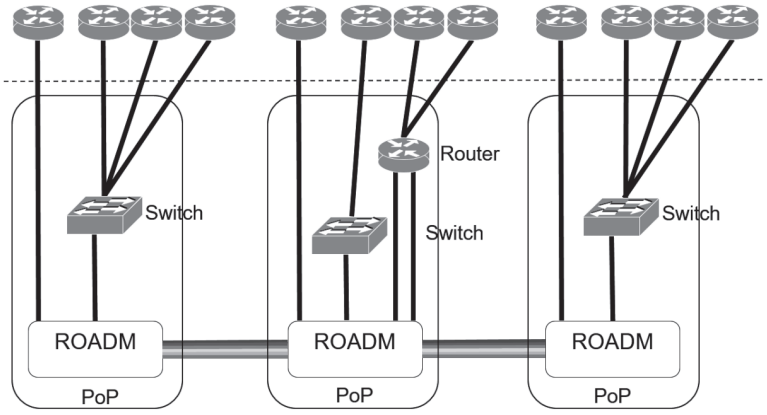


Abb. 4: Aufbau eines Routerless PoP

Wie von Piger [Pi17] dargestellt, bietet dieses Konzept jedoch auch wichtige Vorteile. Zum einen ist in den letzten Jahren ein kontinuierliches Wachstum der globalen IPv4-

Routingtabelle [Hub] zu beobachten, so dass nur noch leistungsfähige Routermodelle die komplette Routingtabelle vorhalten können. Es würde jedoch zu hohen Kosten führen, solche Routermodelle an allen Standorten zu verwenden.

Ein wichtiger Punkt ist auch der Schutz vor DDoS-Angriffen, die hohe Datenraten erzeugen. Durch diese hohen Datenraten wird nicht nur der Zielkunde des Angriffs betroffen, sondern auch viele weitere Kunden. Daher dürfen die Betreiber von Weitverkehrsnetzen die Abwehr von solchen Angriffen nicht ihren Kunden überlassen, sondern müssen in Zusammenarbeit mit den Kunden dafür sorgen, dass solche Angriffe an zentralen, sehr leistungsfähigen Standorten im Weitverkehrsnetz herausgefiltert werden.

4 Redundanzkonzepte

Beim Aufbau eines Weitverkehrsnetzes muss auch entschieden werden, welche Redundanzen man im Netz vorsieht, um für Fehlerfälle vorbereitet zu sein. Die Betrachtung soll hier von unten her, d.h. von der optischen Ebene zur Routerebene erfolgen.

Zunächst einmal ist dafür zu sorgen, dass für jeden PoP mindestens ein zweiter Weg existiert, um diesen zu erreichen. Dabei ist es wichtig, von den Anbietern von Glasfaserstrecken genaue geographische Angaben bis zu den einzelnen Straßen hinab zu verlangen. Ansonsten können sich Situationen ergeben, dass Wege durch die gleichen Straßen führen, obwohl in einer abstrahierten logischen Darstellung keine fehlenden Redundanzen zu erkennen sind.

Wenn man den Kunden dedizierte Wellenlängen anbietet, dann hat man nur eine Möglichkeit zum Schutz, nämlich die *Optical 1+1 Protection*. Das heißt neben der eigentlichen Wellenlänge wird eine zweite Wellenlänge fest geschaltet, die als Backup dient. Die Umschaltung kann im Fall einer Störung automatisch in weniger als 50 ms erfolgen.

Wenn es um die Verbindungen zwischen Routern geht, dann hat man mehr Möglichkeiten. Auch hier kann man eine Optical 1+1 Protection schalten, was aber mit einem hohen Aufwand einhergeht. Stattdessen kann man sich auf die Möglichkeiten des Routings verlassen, d.h. dass der Routing-Algorithmus ggf. einen neuen Weg auswählt. Dieses geht jedoch mit 1 bis 2 s nicht ganz so schnell wie das Umschalten auf optischer Ebene.

Eine weitere Option, die man je nach den Möglichkeiten der DWDM-Plattform noch hat, ist die *Optical Restoration*. In diesem Fall würden bei Störungen dynamisch neue Pfade auf der optischen Ebene gefunden. Dieses dauert jedoch bis zu 30 s und hängt auch davon ab, ob in der konkreten Situation noch alternative Wege existieren, die für dynamisch geschaltete optische Pfade verwendet werden können.

5 Fazit und Ausblick

Wenn man sich die Entwicklung in den letzten Jahren ansieht, dann kann man feststellen, dass viele grundsätzliche Abwägungen auch schon bei früheren Netzgenerationen zu beachten waren. Deutlich verschärft hat sich jedoch die Relevanz von DDoS-Angriffen, wobei man gerade durch viele schlecht gewartete Internet of Things-Geräte in Zukunft eher noch mit einer weiteren Zunahme solcher Angriffe rechnen muss. Dieses muss im Netzdesign berücksichtigt werden, so dass es sinnvoll erscheint, gerade Standorte mit Anbindung nach außen so leistungsfähig auszustatten, dass dort eine wirksame DDoS-Abwehr erreicht werden kann. Definierte Prozesse für die Zusammenarbeit von Netzbetreibern mit betroffenen Kunden sind dabei sehr wichtig.

Abschließend sollen einige aktuelle Entwicklungen betrachtet werden.

5.1 Alien Wavelength

Wenn mit dem DWDM-System eines Herstellers zusätzlich Wellenlängen übertragen werden, die von der optischen Technik eines anderen Herstellers erzeugt werden, dann spricht man von *Alien Wavelength*. Nachdem diese Technik einige Jahre lang in größeren Versuchen untersucht wurde, befindet sich diese nun auch in manchen Szenarien im Produktiveinsatz. Das ist der Fall beim schwedischen Forschungsnetz SUNET, wobei die Hersteller Adva (DWDM-Technik) und Juniper (Router) eng miteinander kooperieren, aber auch beim italienischen Forschungsnetz GARR. GARR setzt Technik von zwei unterschiedlichen DWDM-Hersteller in Nord- und Süditalien ein und verbindet die Systeme über Alien Wavelength. Das tschechische Forschungsnetz CESNET ist in diesem Bereich sehr aktiv und verwendet im eigenen Netz die DWDM-Technik von unterschiedlichen Herstellern und entwickelt auch eigene DWDM-Systeme (siehe Berichte beim CEF-Workshop [Cu] sowie Erfahrungsberichte bei der TNC [TN]).

Auch wenn die Beispiele zeigen, dass die Verwendung von Alien Wavelength möglich ist, stellt sich aber dennoch die Frage, ob diese einen Zusatznutzen bieten. Bei einem klaren Generationenkonzept der DWDM-Technik sind diese eigentlich nicht notwendig. Eventuell könnten sie beim Übergang zwischen Netzgenerationen nützlich sein oder bei der Verbindung zu anderen Netzen.

5.2 Software Defined Networking

Eine andere Frage ist die zukünftige Relevanz von Software Defined Networking für Weitverkehrsnetze. SDN würde in diesem Kontext bedeuten, dass die Hersteller der DWDM- und Router-Technik nur noch Geräte mit einer Grundfunktionalität liefern, während die komplexen Funktionen separat davon in Software programmiert werden. Diese komplexen

Funktionen würden es insbesondere erlauben, die Ebenen (optische Ebene, Switching, Routing) in intelligenter Weise miteinander zu kombinieren, wie man das bisher durch die relativ getrennten Herstellerwelten nicht kann (siehe White Paper von ECI Telecom [Hua]). Der Vorteil besteht also darin, dass man das Management deutlich mehr nach eigenen Vorstellungen und betrieblichen Notwendigkeiten gestalten kann und bei Änderungswünschen nicht mehr auf die Kooperationsbereitschaft des Herstellers angewiesen ist. Bislang waren die Hersteller jedoch nicht bereit davon abzugehen, geschlossene DWDM-Systeme bestehend aus Hardware und Software zu liefern. Allerdings versucht das 2016 gestartete Telecom Infra Project [Te] die Situation aufzubrechen und die Entwicklung offener Systeme zu fördern. Obwohl auch mehr als 500 andere Firmen mitwirken, nehmen gerade Facebook und die Deutsche Telekom eine wesentliche Rolle dabei ein. Sie erhoffen sich dadurch die Eintrittsbarrieren in den DWDM-Markt zu senken, der durch etablierte Anbieter gekennzeichnet ist. Außerdem sollen so Innovationen von Start-Up-Firmen gefördert werden. Im Bereich der Router wird der Einsatz von kleineren Geräten („Pizzabox“) favorisiert, die aber flexibel zu einem großen Router für Kernnetzstandorte kombiniert werden können. Damit erwartet man auch in diesem Bereich von etablierten Großgeräten führender Hersteller unabhängiger zu werden.

Eine wesentliche Frage ist aber wie man die sehr hohen Anforderungen an die Betriebsstabilität des Netzes erfüllt, wenn die komplexen Netzwerkfunktionen von Dritten programmiert werden. Es ist dabei noch nicht absehbar, ob man auf eine Art von Open Source Betriebssoftware setzen kann oder eventuell spezielle Dienstleister braucht, die solche Software liefern.

5.3 Entwicklung des Datenvolumens

Eine längerfristige Herausforderung wird auf Weitverkehrsnetze zukommen, wenn die Bitraten im Access Bereich sehr stark zunehmen. Seit der Einführung der DWDM-Technik um die Jahrtausendwende stellen die Weitverkehrsnetze so gut wie nie den Flaschenhals dar, weil sie durch die Installation von weiteren Wellenlängen auf schon existierenden Glasfaserverbindungen relativ kurzfristig ausgebaut werden können. Würde es aber so sein, dass jedem Endnutzer Datenraten im Gigabit/s-Bereich zur Verfügung stehen würden (z.B. für 4K Videos, Cloud-Nutzung, etc), dann würden sich große Datenvolumina im Weitverkehrsnetz aggregieren, so dass auch die Terabit/s-Kapazitäten der heutigen DWDM-Systeme voll ausgenutzt würden. Dann könnten Kapazitätssteigerungen im Weitverkehrsbereich eventuell tatsächlich weitere Glasfasern erforderlich machen, was deutlich aufwändiger und kostenintensiver ist.

Im Zusammenhang dem Anstieg des Datenvolumens ist auch die Entwicklung des Energieverbrauchs zu betrachten. Hierbei ist festzustellen, dass der Energieverbrauch der optischen Technik im wesentlichen unabhängig davon ist, wie viele Daten tatsächlich übertragen werden. Beim Routing dagegen steigt der Energieverbrauch mit dem Datenvolumen an [Hi11]. Hiermit steigen dann auch die Anforderungen an die Klimatisierung sowie USVs. Es ist

daher aus Sicht von Netzbetreibern sinnvoll, Datenverkehr zwischen zwei Kundenzugängen, der dauerhaft und mit hohem Datenvolumen beobachtet wird, möglichst als dedizierte Wellenlänge oder über eine Aggregationsplattform zu realisieren, um die Routing-Plattform zu entlasten. An dieser Stelle wäre es interessant geeignete Kriterien zu entwickeln, wann ein solcher Offload durchgeführt werden sollte.

Längerfristig wären dann auch Bandwidth-On-Demand-Realisierungen interessant, bei denen Wellenlängen automatisch je nach Bedarf geschaltet werden. An dieser Stelle ist es jedoch wichtig zu erkennen, dass man erst eine kritische Masse an möglichen relevanten Kunden und Infrastruktur braucht. Ansonsten würde mit diesem Konzept oftmals zuviel an Kapazität ungenutzt bleiben.

Literaturverzeichnis

[AGN12] Autenrieth, A.; Gunkel, M.; Neugirg, M.: Cost Savings and Robustness Improvements in IP-over-DWDM Core Networks by Employment of CD-ROADMS and Advanced Multilayer Survivability Schemes. In: ITG-Fachbericht 233: Photonische Netze. VDE, Mai 2012.

[Ak] Akamai Technologies, Q3 2016 State of the Internet Report - Security Report, Nov 2016. https://content.akamai.com/pg7407-soti-security-report-q3-en.html.

[Ci] Cisco: Packet Optical Convergence. https://www.cisco.com/c/en/us/solutions/service-provider/ipodwdm/index.html.

[Cu] Customer Empowered Fibre Networks Workshop, Prague, Sep. 2017. https://www.cesnet.cz/events/cef2017/.

[Gi] Gill, Jay: ICE5: Innovation on Fast Forward, Mar. 2018. https://www.infinera.com/ice5-innovation/.

[Hi11] Hinton, Kerry; Baliga, Jayant; Feng, Michael Z.; Ayre, Robert; Tucker, Rodney S.: Power consumption and energy efficiency in the internet. IEEE Network, 25(2):6–12, 2011.

[Hua] Huma, Jonathan: Trains, planes and multi-layered software defined networks, Spring 2017. http://www.ecitele.com/media/2012/trains-planes-and-more-fibre-systems-spring-2017.pdf.

[Hub] Huston, Geoff: BGP in 2017, Jan. 2018. http://www.potaroo.net/ispcol/2018-01/bgp2017.html.

[Pi17] Piger, S.: Der Erneuerung letzter Akt: Eine neue Aggregationsplattform für das X-WiN. In: DFN-Mitteilungen, Heft 91. DFN, Mai 2017.

[Te] Telecom Infra Project: Open Optical Packet Transport. https://telecominfraproject.com/project-groups-2/backhaul-projects/open-optical-packet-transport/.

[TN] TNC 16, Session, 4D - Alien Wavelengths – from field trials to profitable services, Prague, June, 2016. https://tnc16.geant.org/core/session/73.

Review skalierbarer Netzwerkdesign-Prinzipien zur Optimierung des Campus-Edge für BigData Forschung

Die ScienceDMZ mit Data Transfer Nodes (DTN)

Jakob Tendel [1] [2]

Abstract: Für bandbreitenstarke Datenübertragung ist es erforderlich, dass die Netzwerksysteme bis zu den Endpunkten der Übertragung hin für gute Performance geeignet und konfiguriert sind. Die per TCP Protokoll praktisch zu erreichende Datenrate hat einige Abhängigkeiten mit in Standortnetzwerken auftretenden Faktoren. Dazu gehören Spezifikation und Konfiguration der Endsysteme und Netzwerkkomponenten auf dem Pfad, sowie Quellen von Paketverlust. Der negative Einfluss dieser Faktoren auf den zu erreichenden Durchsatz skaliert nichtlinear mit der Paketlaufzeit, also der Entfernung zwischen den Endpunkten der Verbindung, sodass bei Weitverkehrsverbindungen mitunter gravierende Einbrüche der Datenrate auftreten können. Die koordinierte und konsequente Anwendung einiger gängiger Best-Practices, wie im Folgenden vorgestellt, kann bereits eine erhebliche Durchsatzoptimierung bringen. Im Wesentlichen wird die Entflechtung auf Netzarchitektur-Ebene von bandbreitenstarken BigData Anwendungen und der alltäglichen Nutzung des Campus-Netzwerks empfohlen, damit eingesetzte Komponenten und Architekturen auf die jeweiligen Bedürfnisse optimiert werden können und störende Wechselwirkungen vermieden werden. Die Maßnahmen umschließen die Schaffung eines Hochleistung-Netzsegments mit möglichst direktem Anschluss an den Campusrouter zum Forschungsnetz (der ScienceDMZ) und den Einsatz von optimierten Host-Systemen (Data Transfer Nodes - DTN) in der DMZ. Diese Architektur ist ebenfalls bereits bestens geeignet für den Einsatz zusammen mit software-gesteuerten Forschungsdaten-Portalen oder Research-Gateways mit systematisch automatisiertem Datenaustausch in ortsunabhängigen Forschungsvorhaben.

Keywords: Forschungsnetz, BigData, Paketverlust, ScienceDMZ, DTN.

1 Einleitung

Die Übertragung großer Forschungsdatensätze über große Entfernungen wird für die zunehmend internationalen Forschungsvorhaben immer wichtiger. Beispiele für Datenmengen in Petabyte-Größenordnung sind Klimadaten, Genomdaten, Satellitenbilder, oder Physik-Experimente, welche Forschungsgruppen mitunter über Kontinente hinweg miteinander austauschen. Für praktikabel zeitnahe Anwendungen sind hier dauerhafte Datenraten im Gigabit bzw. multi-10Gigabit Bereich für einzelne Verbindungen („Flows“) notwendig, ein im regulären LAN und Internet-Bereich

[1] DFN-Verein e.V., Alexanderplatz 1, 10178 Berlin, tendel@dfn.de
[2] GÉANT Association

unübliches und daher nicht standardmäßig optimal unterstütztes Verkehrsmuster. Die erreichbaren Datenübertragungsraten solcher Flows sind trotz der hierfür speziell optimierten Forschungsnetze abhängig von Netz-Architekturen und Systemen in Einrichtungen an beiden Enden. Dieser Review-Artikel stellt die grundlegende Problematik des Performanceverlusts vor, geht auf Lösungsansätze wie die ScienceDMZ ein, und stellt einige praktische Umsetzungen vor. Es wird eine Sammlung von Design-Prinzipien und Technologie-Bausteinen vorgestellt, wie Systeme und Netz am Campus-Edge zum Forschungsnetz optimal gestaltet werden können. Ziel ist es, BigData Spitzenforschung und alltägliche Nutzung des Campus-Netzes bei Wahrung der notwendigen IT-Sicherheit in Einklang zu bringen.

2 Problemstellung

Ein möglich auftretender Effekt bei der Übertragung von Daten über große Entfernungen ist eine mit steigender Entfernung stark nachlassende Übertragungsrate, während im lokalen Netzwerk die theoretisch zu erreichende Datenrate meist zufriedenstellend approximiert werden kann. Selbstverständlich liegen diverse mögliche Fehlerquellen und Flaschenhälse auf der Strecke zwischen den Endpunkten, z.B. Überlast oder asymmetrisches Routing, die allesamt die Datenrate stark beeinträchtigen können. Forschungsnetze sind jedoch gezielt optimiert und überwacht, um solche Fehlerquellen zu minimieren.

An dieser Stelle wird konkret auf Störfaktoren rund um die Endpunkte von Flows, also von Verbindungen über das TCP-Protokoll, eingegangen. Die Literatur z.B. [Da13] beschreibt mehrere Fehlerquellen und Einflussgrößen, welche in Kombination die zu beobachtenden Leistungsverluste ergeben. Diese gliedern sich grob in nicht-optimierte Software und Hardware sowie Paketverlust.

2.1 Das TCP Protokoll

Den meisten Verfahren zur Übertragung von Datensätzen über Computer-Netzwerke liegt das TCP Protokoll (engl. "Transmission Control Protocol") zugrunde, essentieller Bestandteil der Familie der Internet-Protokolle zur paketvermittelten Kommunikation. Es stellt mittels automatischer Mechanismen zur Synchronisierung und Bestätigung (Three-Way-Handshake) zwischen zwei Endpunkt-Systemen eine Verbindung für den zuverlässigen Austausch von Datenpaketen her.

2.2 TCP Einstellungen für lange Strecken

TCP hat Mechanismen zur Flusssteuerung und Überlaststeuerung, um möglichst zuverlässigen Datendurchsatz zu ermöglichen. Diese benötigen jedoch für den jeweiligen

Anwendungsbereich adäquate Einstellungen, um nicht selbst zur Durchsatzbremse zu werden. In einer Situation mit langen Umlaufzeiten („Round-Trip-Time“; RTT) ab ca.10-20ms kann eine fehlerfreie Übertragung unter Umständen trotzdem ein timeout und damit eine unnötige wiederholte Übertragung („Retransmission“) veranlassen. Bei bandbreitenstarken Flows mit großen RTT kann das „Bandwidth- Delay- Product“ in Größenordnungen wachsen, die das vereinbarte TCP Window übersteigen und zur Drosselung der Übertragung führen. Mit der TCP Window Size wird von einem Empfänger die maximale Datenmenge angegeben, die ohne Empfangsbestätigung (TCP ACK) verarbeitet wird. Nicht optimal konfigurierte Systeme sind in der Folge selbst bei null Fehlern nicht in der Lage, die volle Leistung des Netzwerks auszunutzen. Trotz der Verbreitung von Jumbo-Frames zur Steigerung der Datenmenge pro Paket, oder der Verfügbarkeit neuerer Steueralgorithmen und Autotuning, erfordert Performance-Tuning eines TCP-Stacks nach wie vor die bewusste Betrachtung des beabsichtigten Einsatzbereichs, um die TCP Parameter mit ihren zahlreichen teils gegensätzlichen Effekten optimal abzustimmen.

2.3 Anforderungen an Router

Um ideale Bedingungen für große Datenraten herzustellen, muss neben den TCP Einstellungen der Hosts an den Endpunkten auch die Netzwerkinfrastruktur auf dem Übertragungspfad den Anforderungen an verlustfreie Übertragung starker Flows gewachsen sein. Das erfordert die Fähigkeit, den Inhalt großer TCP Sende-Puffer mit der vollen Line-Rate eines Host Systems zu verarbeiten. Idealerweise enthält der Pfad durch das Netzwerk lediglich performante Switches und Router, und davon so wenige wie möglich. Jede Netzwerkkomponente stellt eine potentielle Fehlerquelle oder einen Engpass dar. Ganz besonderes Augenmerk gilt der Dimensionierung der Router-Puffer, die selbst bei Konfluenz mehrerer starker Flows auf einem Sende-Interface den Datenfluss bewältigen können müssen. Dies ist häufig bei „kleinen“ LAN Routern und Switches nicht ausreichend gegeben.

2.4 TCP und Paketverlust

Da in der Praxis jedoch keine idealen Bedingungen vorausgesetzt werden können, muss man die Situation inklusive Übertragungsfehler betrachten. Übertragungsfehler an dieser Stelle bedeutet, dass eines der Pakete der TCP Verbindung nicht die Gegenstelle erreicht hat, ein sogenannter Paketverlust. Ein Paketverlust wird durch die ausbleibende Bestätigung schnell erkannt und kann üblicherweise durch eine schnelle Wiederholung der Übertragung (fast retransmit) kompensiert werden. Dies funktioniert jedoch nur bis zu einer gewissen Paketverlustrate, ab der dann der TCP Algorithmus die gesamte Übertragungsrate herunter regelt. Je nach eingestelltem Algorithmus fallen diese Drosselungen teils drastisch aus und halten die Übertragungsrate lange Zeit unterhalb des Idealwertes. Die Abhängigkeit dieser Prozesse von der RTT bedeuten, dass die kritische Paketverlustrate für eine optimale Übertragungsgeschwindigkeit also mit steigender RTT

sinkt, bzw. bei konstanter Paketverlustrate sinkt die mögliche Übertragungsgeschwindigkeit mit der RTT (Abb. 1). Im Zusammenspiel mit suboptimal konfigurierten Systemen verursacht Paketverlust je nach RTT der Verbindung massive Einbrüche in der Übertragungsrate. Durch die physikalischen Gegebenheiten steigt die Signallaufzeit und damit auch die RTT linear mit der Entfernung (z.B. Glasfaser addiert ca. 2x1ms pro 100km zusätzlich zur RTT, hin/rück).

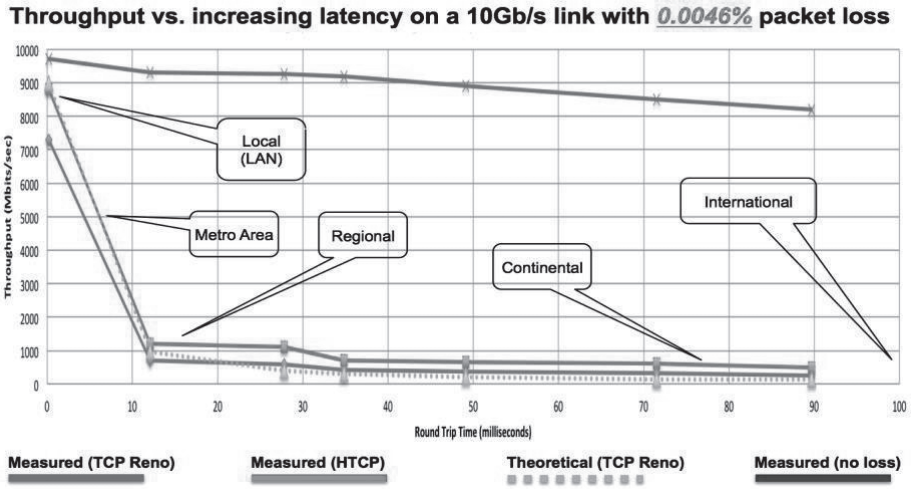


Abb. 1: Datendurchsatz vs RTT [fasterdata.es.net]

2.5 Ursachen für Paketverlust

Mit der Digitalisierung aller Facetten des Betriebs an Einrichtungen bauen zahllose andere Anwendungen nun ebenfalls auf die Infrastruktur. Viele dieser Anwendungen, wie für Personalangelegenheiten oder Finanzen mit erhöhtem Schutzbedarf, und der Einsatz von Standard-PC Systemen für Mitarbeiter und Studenten, haben die Anwendung diverser Komponenten und Praktiken für IT-Sicherheit und -Management im Campusnetz notwendig gemacht, die in der reinen Forschungsdaten-Verarbeitung unnötig oder gar hinderlich sind. Dazu gehören Firewalls, Intrusion Detection Systeme, VPN Gateways, Limiter, Proxys, NAT-Gateways, etc. Alle zusätzlichen aktiven Netz-Komponenten, besonders solche die Deep-Packet-Inspection (DPI) durchführen, können stark limitierende Flaschenhälse für Flows von Wissenschaftsdaten sein. Sie können in vielen Fällen auch den so schädlichen Paketverlust verursachen, da sie einzelne extrem starke Flows nicht in voller Geschwindigkeit verarbeiten können. Dazu kommt, dass DPI auf Wissenschaftsdaten in den meisten Fällen wenig sinnvoll ist, da es sich hier um große Sätze Binärdaten handelt und nicht um Webinhalte mit potentiellen Schadskripten oder persönlichen Daten.

3 Lösungsansatz

Die vorgestellten Ansätze verfolgen allesamt das Ziel, eine dynamische Forschungstätigkeit mit großen Datenmengen über verteilte Standorte zu ermöglichen, und dabei gegenseitig schädliche Wechselwirkungen mit der alltäglichen Netz/Internet-Nutzung auf einem Campus zu vermeiden. Dies geschieht im Wesentlichen durch eine infrastrukturelle Trennung der Datenpfade vom allgemeinen Campusnetz. Zur Vermeidung multipler hops durch ein hoch ausgelastetes Campusnetz voll heterogener Anwendungen und Infrastruktur, sehen diese Ansätze, analog einer Webserver-DMZ, die Einrichtung einer spezialisierten Netz-Enklave neben dem Übergang zum Forschungsnetz vor. Dort befinden sich dedizierte und optimierte Übertragungs-Server, welche dann ungestört die Fernübertragung der Forschungsdaten übernehmen. Dabei wird der Sicherheit im erforderlichen Umfang Rechnung getragen, jedoch mit anderen Methoden und Metriken als in der Unternehmens-IT Praxis.

3.1 Die Science DMZ

Entwickelt am „Energy Sciences Network“ ESnet des US Energieministeriums DoE steht ScienceDMZ für ein bereits mehrfach bewährtes Design-Muster für Campus-Infrastruktur an Einrichtungen mit Bedarf an schnellem Austausch von Forschungsdaten. Dazu gehören die dem DoE unterstehenden und von ESnet betreuten nationalen Forschungslabore der USA, mit Physik-, Material-, und HPC-Forschung an weit verteilten Standorten, aber auch die Universitäten und andere Einrichtungen, an denen mit solchen Daten geforscht wird. Umfassend beschrieben in [Da13] und seither vielfältig zitiert und implementiert [Ma14], [Pe17], stellt die ScienceDMZ einen Baukasten von Best Practices zur Mitigation der eingangs beschriebenen Problemstellung dar. Die ScienceDMZ ist ein generalisierter Satz von Design-Mustern, der flexibel und skalierbar je nach den lokalen Gegebenheiten und Bedarfen eine passende Lösung ermöglicht.

Die Science DMZ bietet:

- Eine skalierbare und erweiterbare Netzwerkplattform ohne Paketverlust, speziell optimiert für die Übertragung umfangreicher Wissenschaftsdaten
- Dem tatsächlich notwendigen Sicherheitsniveau angemessene Nutzungsrichtlinien, damit performante Anwendungen nicht unnötig eingeschränkt werden
- Eine effektive Anbindung lokaler Ressourcen an die Weitverkehrsnetze
- Mechanismen zur laufenden Messung der Netzperformance

Die einfachste Ausführung der ScienceDMZ (Abb. 2) besteht aus einem separaten Netzbereich außerhalb der Campus Firewall, angeschlossen an den Border-Router durch einen dedizierten DMZ Router. In der DMZ befindet sich ein spezialisierter Server (Data

Transfer Node – DTN) zur Datenübertragung, sowie eine perfSONAR[3] Box zur Messung der Netzperformance. Die DTN kann direkt über nur zwei Router mit dem WAN kommunizieren, ohne den Einfluss des weiteren Campus-LAN. Die Forscher im Campus LAN haben über den kurzen Pfad zur DTN wegen der niedrigen Latenz kaum Performance Einbußen zu befürchten.

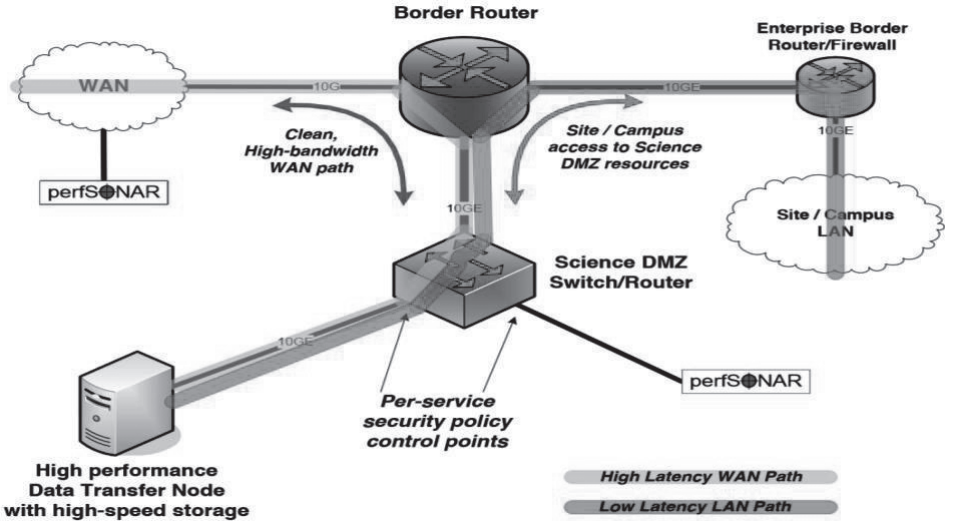


Abb. 2: Eine einfache ScienceDMZ [Da13]

Traditionelle Werkzeuge der Netzwerksicherheit, wie Firewalls und andere Middleboxen, erzielen ihre Wirkung größtenteils durch die Analyse von Web-Traffic auf aktive Schad-Inhalte. In diesem Kontext werden aber lediglich große Datensätze ohne aktive Inhalte und über sehr wenige Anwendungen und Ports bewegt. Aktive Analyse bringt hier also wenig tatsächlichen Sicherheits-Mehrwert und kann wie beschrieben darüber hinaus schädlich auf die Datenrate wirken. Deswegen wird die Sicherheit der DMZ ohne Einsatz aktiver Analyse mit mehreren ineinandergreifenden Maßnahmen realisiert. Zum einen bedarf es geeigneter Security Policies auf dem DMZ Router, welcher unter Anderem als statische Paketfilter-Firewall mit voller Leitungsgeschwindigkeit fungiert und über Access Control Lists (ACL) zur Begrenzung der Dienste auf spezifisch benannte Gegenstellen verfügt. Zum anderen sind die DTN Systeme gehärtet und nicht zur interaktiven Bedienung durch Anwender konfiguriert. So arbeiten sie als „headless“ Systeme mit stark eingeschränktem Funktions-Umfang an nach außen hin offenen Schnittstellen und Protokollen.

[3] https://www.perfsonar.net/

Durch diese Entflechtung der Netzinfrastruktur profitiert auch der Rest des Campusnetzwerks von der reduzierten Beanspruchung. Die IT-Sicherheit im restlichen Campus kann erhöht werden, weil Sonderregeln und Löcher in der Firewall zurückgenommen werden können und sich insgesamt die Angriffsfläche reduziert. So wird dem Bedarf der wissenschaftlichen Anwendungen an Performance gerecht, ohne die Sicherheit zu vernachlässigen.

Eine Eigenschaft der ScienceDMZ ist, dass sie fast beliebig skaliert werden kann. Am Beispiel einer HPC Einrichtung zeigt [Da13] die Möglichkeit der Parallelisierung vieler Netzkomponenten und Pfade, um Lastverteilung und Ausfallsicherheit zu gewährleisten. Mehrere DTN übernehmen die Datenübertragung und können dabei auf ein mit dem Cluster gemeinsames Dateisystem zugreifen. So werden gar keine großen Datenströme ins Campusnetz geführt, sondern verbleiben gleich im Kontext der HPC Anlage. Ebenfalls werden so die Login Knoten des HPC Systems von der Datenübertragung entlastet.

Je komplexer die Auslegung der DMZ wird, desto wichtiger ist die Überwachung der Netzperformance auch in Teilsegmenten. Man hat in diesem Fall also mehrere perfSONAR Boxen an strategischen Punkten auf dem Datenpfad, um die kritischen Teilsegmente DTN-Router und Router-WAN separat ausmessen zu können.

3.2 Die Data Transfer Node

Um unter den beschriebenen Bedingungen dauerhaft performante TCP Verbindungen aufrecht zu erhalten, hat sich der Einsatz dedizierter und optimierter Host Hardware bewährt. Unter dem Begriff Data Transfer Node – DTN[4] wird eine Refenrenz-Konfiguration mit Hardware der Server-Klasse und einer speziell zusammengestellten Software-Konfiguration auf Linux-Basis vorgestellt. Eine Priorität sind selbstverständlich Netzwerk-Interfaces höchster Qualität. Zugang zu schnellem Massenspeicher ist ebenso notwendig, um die Übertragung nicht durch I/O Limits auszubremsen. Software-seitig kommen gängige Datentransfer-Tools wie GridFTP, GLOBUS online, oder SSH/SCP mit high-performance patches sowie ein getunter TCP Stack zum Einsatz. Die Konfiguration ist im Hinblick auf laufende Services und offene Ports stark eingeschränkt und gehärtet, um die Stabilität und Angriffsfläche zu optimieren. Die Empfehlungen gehen bis hin zu Treiber-Versionen und Zuweisung einzelner Prozesse zu CPU Kernen, was nachweislich Einfluss auf Stabilität und Durchsatz hat.

[4] http://fasterdata.es.net/science-dmz/DTN/

3.3 Research Data Portal

Es werden zunehmend integrierte und effiziente Umgebungen und IT Infrastrukturen für datenintensive Forschung an verteilten Standorten nachgefragt. Diese Forderung nach Integration beinhaltet zunehmend auch geeignete Softwaresysteme zur Datenlogistik und Bearbeitung. Diese "Research Data Portal" genannten Dienste basieren traditionell auf Webservern, was jedoch an Grenzen der Skalierbarkeit und Flexibilität stößt. Aufbauend auf der ScienceDMZ haben [Ch18] eine moderne Fassung entworfen, welche die vormals im Server vereinten Funktionen der Anfragesteuerung, Datenübertragung, und Autorisation/Koordination aufteilen. Der Server im Campusnetz bleibt erhalten, übernimmt aber nur noch die klassischen Webserver-Aufgaben wie z.B. Das Benutzer-Interface und die Suchfunktion. Die Datenübertragung erfolgt durch DTNs und die Koordination und Zugangssteuerung des ganzen kann mit einem externen Data Manager wie Globus [Ch14] durchgeführt werden. Mit einer ScienceDMZ ist eine Einrichtung also bestens auf die Unterstützung moderner Forschungsdaten-Portale vorbereitet.

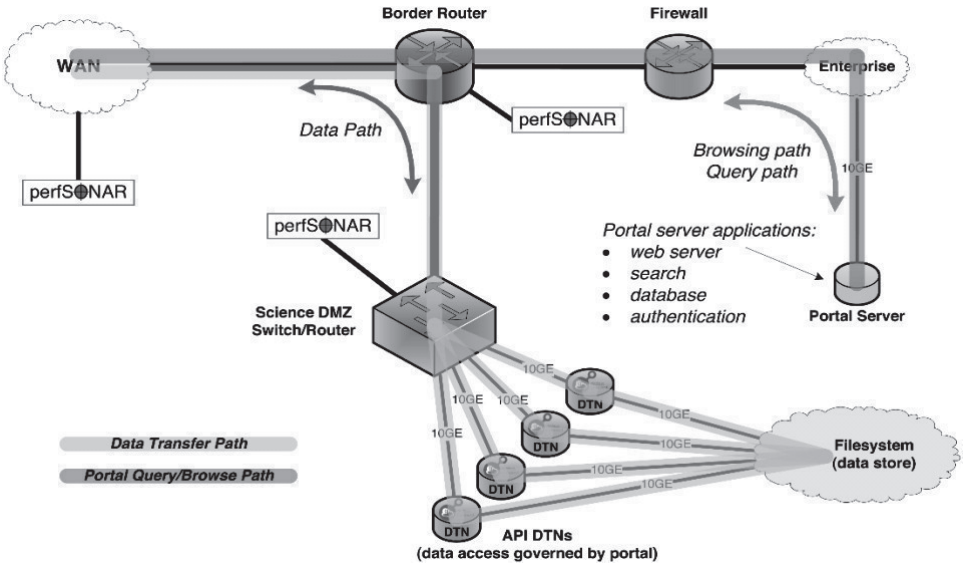


Abb. 3: Modern Research Data Portal [Ch18]

3.4 Fallbeispiel University Otago, Neuseeland

Die University of Otago in Neuseeland, steht vor der Herausforderung, dass alle internationalen Verbindungen automatisch über sehr große Entfernungen gehen. Eine optimale Ausnutzung der Dateninfrastruktur ist also essentiell.

Forscher stellten eine ernüchternd niedrige maximale Datenrate im Bereich weniger 100Mbps aus dem Campus-Netz zum nationalen Forschungsdatenspeicher und zu internationalen Partnern fest. In Zusammenarbeit mit Ihrem Forschungsnetz REANNZ richtete die Universität Otago eine ScienceDMZ ein[5] und konnte so eine dramatische Optimierung der maximalen Datenübertragungsrate erzielen [Abb. 4].

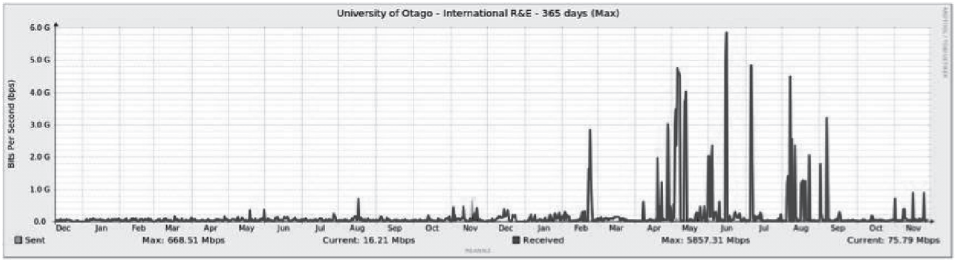


Abb. 4: Erreichbare Lastspitzen vor und nach Einrichtung der ScienceDMZ [REANNZ]

4 Zusammenfassung

In der heutigen Zeit der BigData Forschung ist eine effektive Ausnutzung der vorhandenen Infrastruktur mindestens genau so wichtig wie ständige Leistungssteigerungen. Das Ziel muss sein, die optimale Leistungsfähigkeit der Infrastruktur für die anspruchsvollsten Anwendungen aus der Wissenschaft spürbar und praktisch nutzbar zu machen, ohne dadurch den täglichen Betrieb der Brot-und-Butter Anwendungen im Campus zu beeinträchtigen. Der Trend geht klar in Richtung Entflechtung dieser zwei unterschiedlichen Anwendungsbereiche sowohl bei der Netzarchitektur, als auch der Benutzer-Interfaces und Softwareschnittstellen. So kann die intensive Optimierung für Übertragungsleistung auf einen überschaubaren und schützbaren Netzbereich beschränkt werden. Die vorgestellten Best Practices beschreiben eine moderne Basis für zukünftige Forschungsdaten-Infrastruktur.

Die ScienceDMZ ist natürlich nicht für alle Anwendungen das perfekte Werkzeug. Kritisch zu betrachten ist sicherlich die Umgehung der traditionellen IT Sicherheit. Ein Anschluss von Netzwerkinfrastruktur an Forschungsnetz/Internet ohne Einsatz einer Firewall bedarf einer genaueren Untersuchung der Wirksamkeit der alternativen

[5] https://reannz.co.nz/case-studies/getting-up-to-speed/

Sicherheitsmaßnahmen und der zu übertragenden Datentypen in Bezug auf Schutzklassen. Eine differenzierte Abwägung des Risiko-Nutzen-Verhältnisses ist in jedem Fall durchzuführen.

Die größere Herausforderung bei der Umsetzung ist meist nicht technischer, sondern organisatorischer Natur. Die Notwendigkeit für neue Sicherheitskonzepte und Metriken erfordert die konstruktive und lösungsorientierte Zusammenarbeit unterschiedlicher Funktionsrollen wie Netz-Architektur, IT-Sicherheit und Datenschutz in ungewohnten Konstellationen. Die Erfahrung vieler Einrichtungen zeigt jedoch auch, dass dieser Aufwand die Dienstqualität sowohl für die Wissenschaft als auch für die alltäglichen Nutzer des Campusnetzes spürbar verbessert und neue Möglichkeiten schafft.

5 Literaturverzeichnis

[Ch14] Chard, K. , Tuecke, S., Foster, I.: Efficient and Secure Transfer, Synchronization, and Sharing of Big Data In IEEE Cloud Computing, vol. 1, no. 3, pp. 46-55, Sept. 2014., doi: 10.1109/MCC.2014.52

[Ch18] Chard K, et al.: The Modern Research Data Portal: a design pattern for networked, data-intensive science., 2018, PeerJ Computer Science 4:e144 doi: 10.7717/peerj-cs.144

[Da13] Dart, E., et al: The Science DMZ: A network design pattern for data-intensive science, 2013 SC - International Conference for High Performance Computing, Networking, Storage and Analysis (SC), Denver, CO, 2013, pp. 1-10. doi: 10.1145/2503210.2503245

[Ma14] Magri, D. R. C, et al.: Science DMZ: Support for e-Science in Brazil, 2014 IEEE 10th International Conference on e-Science, Sao Paulo, 2014, pp. 75-78., doi: 10.1109/eScience.2014.53

[Mi15] Miteff, S., Hazelhurst, S.: NFShunt: A Linux firewall with OpenFlow-enabled hardware bypass, 2015 IEEE Conference on Network Function Virtualization and Software Defined Network (NFV-SDN), San Francisco, CA, 2015, pp. 100-106. doi: 10.1109/NFV-SDN.2015.7387413

[Pe17] Peisert,S., et al: The medical science DMZ: a network design pattern for data-intensive medical science, Journal of the American Medical Informatics Association, , ocx104, doi: 10.1093/jamia/ocx104

ESnet Fasterdata Knowledge Base http://fasterdata.es.net/, 12.2.2018

Petascale DTN Project, https://cs.lbl.gov/news-media/news/2017/esnets-petascale-dtn-project-speeds-up-data-transfers-between-leading-hpc-centers/, 12.2.2018

SWITCH Stories: Wissen verwalten im Zeitalter von Big Data, https://www.switch.ch/de/stories/big_science_data/, 13.2.2018

GI-Edition Lecture Notes in Informatics

P-1 Gregor Engels, Andreas Oberweis, Albert Zündorf (Hrsg.): Modellierung 2001.

P-2 Mikhail Godlevsky, Heinrich C. Mayr (Hrsg.): Information Systems Technology and its Applications, ISTA'2001.

P-3 Ana M. Moreno, Reind P. van de Riet (Hrsg.): Applications of Natural Language to Information Systems, NLDB'2001.

P-4 H. Wörn, J. Mühling, C. Vahl, H.-P. Meinzer (Hrsg.): Rechner- und sensorgestützte Chirurgie; Workshop des SFB 414.

P-5 Andy Schürr (Hg.): OMER – Object-Oriented Modeling of Embedded Real-Time Systems.

P-6 Hans-Jürgen Appelrath, Rolf Beyer, Uwe Marquardt, Heinrich C. Mayr, Claudia Steinberger (Hrsg.): Unternehmen Hochschule, UH'2001.

P-7 Andy Evans, Robert France, Ana Moreira, Bernhard Rumpe (Hrsg.): Practical UML-Based Rigorous Development Methods – Countering or Integrating the extremists, pUML'2001.

P-8 Reinhard Keil-Slawik, Johannes Magenheim (Hrsg.): Informatikunterricht und Medienbildung, INFOS'2001.

P-9 Jan von Knop, Wilhelm Haverkamp (Hrsg.): Innovative Anwendungen in Kommunikationsnetzen, 15. DFN Arbeitstagung.

P-10 Mirjam Minor, Steffen Staab (Hrsg.): 1st German Workshop on Experience Management: Sharing Experiences about the Sharing Experience.

P-11 Michael Weber, Frank Kargl (Hrsg.): Mobile Ad-Hoc Netzwerke, WMAN 2002.

P-12 Martin Glinz, Günther Müller-Luschnat (Hrsg.): Modellierung 2002.

P-13 Jan von Knop, Peter Schirmbacher and Viljan Mahni_ (Hrsg.): The Changing Universities – The Role of Technology.

P-14 Robert Tolksdorf, Rainer Eckstein (Hrsg.): XML-Technologien für das Semantic Web – XSW 2002.

P-15 Hans-Bernd Bludau, Andreas Koop (Hrsg.): Mobile Computing in Medicine.

P-16 J. Felix Hampe, Gerhard Schwabe (Hrsg.): Mobile and Collaborative Business 2002.

P-17 Jan von Knop, Wilhelm Haverkamp (Hrsg.): Zukunft der Netze –Die Verletzbarkeit meistern, 16. DFN Arbeitstagung.

P-18 Elmar J. Sinz, Markus Plaha (Hrsg.): Modellierung betrieblicher Informationssysteme – MobIS 2002.

P-19 Sigrid Schubert, Bernd Reusch, Norbert Jesse (Hrsg.): Informatik bewegt – Informatik 2002 – 32. Jahrestagung der Gesellschaft für Informatik e.V. (GI) 30.Sept.-3. Okt. 2002 in Dortmund.

P-20 Sigrid Schubert, Bernd Reusch, Norbert Jesse (Hrsg.): Informatik bewegt – Informatik 2002 – 32. Jahrestagung der Gesellschaft für Informatik e.V. (GI) 30.Sept.-3. Okt. 2002 in Dortmund (Ergänzungsband).

P-21 Jörg Desel, Mathias Weske (Hrsg.): Promise 2002: Prozessorientierte Methoden und Werkzeuge für die Entwicklung von Informationssystemen.

P-22 Sigrid Schubert, Johannes Magenheim, Peter Hubwieser, Torsten Brinda (Hrsg.): Forschungsbeiträge zur "Didaktik der Informatik" – Theorie, Praxis, Evaluation.

P-23 Thorsten Spitta, Jens Borchers, Harry M. Sneed (Hrsg.): Software Management 2002 – Fortschritt durch Beständigkeit

P-24 Rainer Eckstein, Robert Tolksdorf (Hrsg.): XMIDX 2003 – XML-Technologien für Middleware – Middleware für XML-Anwendungen

P-25 Key Pousttchi, Klaus Turowski (Hrsg.): Mobile Commerce – Anwendungen und Perspektiven – 3. Workshop Mobile Commerce, Universität Augsburg, 04.02.2003

P-26 Gerhard Weikum, Harald Schöning, Erhard Rahm (Hrsg.): BTW 2003: Datenbanksysteme für Business, Technologie und Web

P-27 Michael Kroll, Hans-Gerd Lipinski, Kay Melzer (Hrsg.): Mobiles Computing in der Medizin

P-28 Ulrich Reimer, Andreas Abecker, Steffen Staab, Gerd Stumme (Hrsg.): WM 2003: Professionelles Wissensmanagement – Erfahrungen und Visionen

P-29 Antje Düsterhöft, Bernhard Thalheim (Eds.): NLDB'2003: Natural Language Processing and Information Systems

P-30 Mikhail Godlevsky, Stephen Liddle, Heinrich C. Mayr (Eds.): Information Systems Technology and its Applications

P-31 Arslan Brömme, Christoph Busch (Eds.): BIOSIG 2003: Biometrics and Electronic Signatures

P-32 Peter Hubwieser (Hrsg.): Informatische Fachkonzepte im Unterricht – INFOS 2003

P-33 Andreas Geyer-Schulz, Alfred Taudes (Hrsg.): Informationswirtschaft: Ein Sektor mit Zukunft

P-34 Klaus Dittrich, Wolfgang König, Andreas Oberweis, Kai Rannenberg, Wolfgang Wahlster (Hrsg.): Informatik 2003 – Innovative Informatikanwendungen (Band 1)

P-35 Klaus Dittrich, Wolfgang König, Andreas Oberweis, Kai Rannenberg, Wolfgang Wahlster (Hrsg.): Informatik 2003 – Innovative Informatikanwendungen (Band 2)

P-36 Rüdiger Grimm, Hubert B. Keller, Kai Rannenberg (Hrsg.): Informatik 2003 – Mit Sicherheit Informatik

P-37 Arndt Bode, Jörg Desel, Sabine Rathmayer, Martin Wessner (Hrsg.): DeLFI 2003: e-Learning Fachtagung Informatik

P-38 E.J. Sinz, M. Plaha, P. Neckel (Hrsg.): Modellierung betrieblicher Informationssysteme – MobIS 2003

P-39 Jens Nedon, Sandra Frings, Oliver Göbel (Hrsg.): IT-Incident Management & IT-Forensics – IMF 2003

P-40 Michael Rebstock (Hrsg.): Modellierung betrieblicher Informationssysteme – MobIS 2004

P-41 Uwe Brinkschulte, Jürgen Becker, Dietmar Fey, Karl-Erwin Großpietsch, Christian Hochberger, Erik Maehle, Thomas Runkler (Edts.): ARCS 2004 – Organic and Pervasive Computing

P-42 Key Pousttchi, Klaus Turowski (Hrsg.): Mobile Economy – Transaktionen und Prozesse, Anwendungen und Dienste

P-43 Birgitta König-Ries, Michael Klein, Philipp Obreiter (Hrsg.): Persistance, Scalability, Transactions – Database Mechanisms for Mobile Applications

P-44 Jan von Knop, Wilhelm Haverkamp, Eike Jessen (Hrsg.): Security, E-Learning. E-Services

P-45 Bernhard Rumpe, Wofgang Hesse (Hrsg.): Modellierung 2004

P-46 Ulrich Flegel, Michael Meier (Hrsg.): Detection of Intrusions of Malware & Vulnerability Assessment

P-47 Alexander Prosser, Robert Krimmer (Hrsg.): Electronic Voting in Europe – Technology, Law, Politics and Society

P-48 Anatoly Doroshenko, Terry Halpin, Stephen W. Liddle, Heinrich C. Mayr (Hrsg.): Information Systems Technology and its Applications

P-49 G. Schiefer, P. Wagner, M. Morgenstern, U. Rickert (Hrsg.): Integration und Datensicherheit – Anforderungen, Konflikte und Perspektiven

P-50 Peter Dadam, Manfred Reichert (Hrsg.): INFORMATIK 2004 – Informatik verbindet (Band 1) Beiträge der 34. Jahrestagung der Gesellschaft für Informatik e.V. (GI), 20.-24. September 2004 in Ulm

P-51 Peter Dadam, Manfred Reichert (Hrsg.): INFORMATIK 2004 – Informatik verbindet (Band 2) Beiträge der 34. Jahrestagung der Gesellschaft für Informatik e.V. (GI), 20.-24. September 2004 in Ulm

P-52 Gregor Engels, Silke Seehusen (Hrsg.): DELFI 2004 – Tagungsband der 2. e-Learning Fachtagung Informatik

P-53 Robert Giegerich, Jens Stoye (Hrsg.): German Conference on Bioinformatics – GCB 2004

P-54 Jens Borchers, Ralf Kneuper (Hrsg.): Softwaremanagement 2004 – Outsourcing und Integration

P-55 Jan von Knop, Wilhelm Haverkamp, Eike Jessen (Hrsg.): E-Science und Grid Adhoc-Netze Medienintegration

P-56 Fernand Feltz, Andreas Oberweis, Benoit Otjacques (Hrsg.): EMISA 2004 – Informationssysteme im E-Business und E-Government

P-57 Klaus Turowski (Hrsg.): Architekturen, Komponenten, Anwendungen

P-58 Sami Beydeda, Volker Gruhn, Johannes Mayer, Ralf Reussner, Franz Schweiggert (Hrsg.): Testing of Component-Based Systems and Software Quality

P-59 J. Felix Hampe, Franz Lehner, Key Pousttchi, Kai Ranneberg, Klaus Turowski (Hrsg.): Mobile Business – Processes, Platforms, Payments

P-60 Steffen Friedrich (Hrsg.): Unterrichtskonzepte für inforrmatische Bildung

P-61 Paul Müller, Reinhard Gotzhein, Jens B. Schmitt (Hrsg.): Kommunikation in verteilten Systemen

P-62 Federrath, Hannes (Hrsg.): „Sicherheit 2005“ – Sicherheit – Schutz und Zuverlässigkeit

P-63 Roland Kaschek, Heinrich C. Mayr, Stephen Liddle (Hrsg.): Information Systems – Technology and ist Applications

P-64 Peter Liggesmeyer, Klaus Pohl, Michael Goedicke (Hrsg.): Software Engineering 2005

P-65 Gottfried Vossen, Frank Leymann, Peter Lockemann, Wolffried Stucky (Hrsg.): Datenbanksysteme in Business, Technologie und Web

P-66 Jörg M. Haake, Ulrike Lucke, Djamshid Tavangarian (Hrsg.): DeLFI 2005: 3. deutsche e-Learning Fachtagung Informatik

P-67 Armin B. Cremers, Rainer Manthey, Peter Martini, Volker Steinhage (Hrsg.): INFORMATIK 2005 – Informatik LIVE (Band 1)

P-68 Armin B. Cremers, Rainer Manthey, Peter Martini, Volker Steinhage (Hrsg.): INFORMATIK 2005 – Informatik LIVE (Band 2)

P-69 Robert Hirschfeld, Ryszard Kowalcyk, Andreas Polze, Matthias Weske (Hrsg.): NODe 2005, GSEM 2005

P-70 Klaus Turowski, Johannes-Maria Zaha (Hrsg.): Component-oriented Enterprise Application (COAE 2005)

P-71 Andrew Torda, Stefan Kurz, Matthias Rarey (Hrsg.): German Conference on Bioinformatics 2005

P-72 Klaus P. Jantke, Klaus-Peter Fähnrich, Wolfgang S. Wittig (Hrsg.): Marktplatz Internet: Von e-Learning bis e-Payment

P-73 Jan von Knop, Wilhelm Haverkamp, Eike Jessen (Hrsg.): “Heute schon das Morgen sehen“

P-74 Christopher Wolf, Stefan Lucks, Po-Wah Yau (Hrsg.): WEWoRC 2005 – Western European Workshop on Research in Cryptology

P-75 Jörg Desel, Ulrich Frank (Hrsg.): Enterprise Modelling and Information Systems Architecture

P-76 Thomas Kirste, Birgitta König-Riess, Key Pousttchi, Klaus Turowski (Hrsg.): Mobile Informationssysteme – Potentiale, Hindernisse, Einsatz

P-77 Jana Dittmann (Hrsg.): SICHERHEIT 2006

P-78 K.-O. Wenkel, P. Wagner, M. Morgenstern, K. Luzi, P. Eisermann (Hrsg.): Land- und Ernährungswirtschaft im Wandel

P-79 Bettina Biel, Matthias Book, Volker Gruhn (Hrsg.): Softwareengineering 2006

P-80 Mareike Schoop, Christian Huemer, Michael Rebstock, Martin Bichler (Hrsg.): Service-Oriented Electronic Commerce

P-81 Wolfgang Karl, Jürgen Becker, Karl-Erwin Großpietsch, Christian Hochberger, Erik Maehle (Hrsg.): ARCS´06

P-82 Heinrich C. Mayr, Ruth Breu (Hrsg.): Modellierung 2006

P-83 Daniel Huson, Oliver Kohlbacher, Andrei Lupas, Kay Nieselt and Andreas Zell (eds.): German Conference on Bioinformatics

P-84 Dimitris Karagiannis, Heinrich C. Mayr, (Hrsg.): Information Systems Technology and its Applications

P-85 Witold Abramowicz, Heinrich C. Mayr, (Hrsg.): Business Information Systems

P-86 Robert Krimmer (Ed.): Electronic Voting 2006

P-87 Max Mühlhäuser, Guido Rößling, Ralf Steinmetz (Hrsg.): DELFI 2006: 4. e-Learning Fachtagung Informatik

P-88 Robert Hirschfeld, Andreas Polze, Ryszard Kowalczyk (Hrsg.): NODe 2006, GSEM 2006

P-90 Joachim Schelp, Robert Winter, Ulrich Frank, Bodo Rieger, Klaus Turowski (Hrsg.): Integration, Informationslogistik und Architektur

P-91 Henrik Stormer, Andreas Meier, Michael Schumacher (Eds.): European Conference on eHealth 2006

P-92 Fernand Feltz, Benoît Otjacques, Andreas Oberweis, Nicolas Poussing (Eds.): AIM 2006

P-93 Christian Hochberger, Rüdiger Liskowsky (Eds.): INFORMATIK 2006 – Informatik für Menschen, Band 1

P-94 Christian Hochberger, Rüdiger Liskowsky (Eds.): INFORMATIK 2006 – Informatik für Menschen, Band 2

P-95 Matthias Weske, Markus Nüttgens (Eds.): EMISA 2005: Methoden, Konzepte und Technologien für die Entwicklung von dienstbasierten Informationssystemen

P-96 Saartje Brockmans, Jürgen Jung, York Sure (Eds.): Meta-Modelling and Ontologies

P-97 Oliver Göbel, Dirk Schadt, Sandra Frings, Hardo Hase, Detlef Günther, Jens Nedon (Eds.): IT-Incident Mangament & IT-Forensics – IMF 2006

P-98 Hans Brandt-Pook, Werner Simonsmeier und Thorsten Spitta (Hrsg.): Beratung in der Softwareentwicklung – Modelle, Methoden, Best Practices

P-99 Andreas Schwill, Carsten Schulte, Marco Thomas (Hrsg.): Didaktik der Informatik

P-100 Peter Forbrig, Günter Siegel, Markus Schneider (Hrsg.): HDI 2006: Hochschuldidaktik der Informatik

P-101 Stefan Böttinger, Ludwig Theuvsen, Susanne Rank, Marlies Morgenstern (Hrsg.): Agrarinformatik im Spannungsfeld zwischen Regionalisierung und globalen Wertschöpfungsketten

P-102 Otto Spaniol (Eds.): Mobile Services and Personalized Environments

P-103 Alfons Kemper, Harald Schöning, Thomas Rose, Matthias Jarke, Thomas Seidl, Christoph Quix, Christoph Brochhaus (Hrsg.): Datenbanksysteme in Business, Technologie und Web (BTW 2007)

P-104 Birgitta König-Ries, Franz Lehner, Rainer Malaka, Can Türker (Hrsg.) MMS 2007: Mobilität und mobile Informationssysteme

P-105 Wolf-Gideon Bleek, Jörg Raasch, Heinz Züllighoven (Hrsg.) Software Engineering 2007

P-106 Wolf-Gideon Bleek, Henning Schwentner, Heinz Züllighoven (Hrsg.) Software Engineering 2007 – Beiträge zu den Workshops

P-107 Heinrich C. Mayr, Dimitris Karagiannis (eds.) Information Systems Technology and its Applications

P-108 Arslan Brömme, Christoph Busch, Detlef Hühnlein (eds.) BIOSIG 2007: Biometrics and Electronic Signatures

P-109 Rainer Koschke, Otthein Herzog, Karl-Heinz Rödiger, Marc Ronthaler (Hrsg.) INFORMATIK 2007 Informatik trifft Logistik Band 1

P-110 Rainer Koschke, Otthein Herzog, Karl-Heinz Rödiger, Marc Ronthaler (Hrsg.) INFORMATIK 2007 Informatik trifft Logistik Band 2

P-111 Christian Eibl, Johannes Magenheim, Sigrid Schubert, Martin Wessner (Hrsg.) DeLFI 2007: 5. e-Learning Fachtagung Informatik

P-112 Sigrid Schubert (Hrsg.) Didaktik der Informatik in Theorie und Praxis

P-113 Sören Auer, Christian Bizer, Claudia Müller, Anna V. Zhdanova (Eds.) The Social Semantic Web 2007 Proceedings of the 1st Conference on Social Semantic Web (CSSW)

P-114 Sandra Frings, Oliver Göbel, Detlef Günther, Hardo G. Hase, Jens Nedon, Dirk Schadt, Arslan Brömme (Eds.) IMF2007 IT-incident management & IT-forensics Proceedings of the 3rd International Conference on IT-Incident Management & IT-Forensics

P-115 Claudia Falter, Alexander Schliep, Joachim Selbig, Martin Vingron and Dirk Walther (Eds.) German conference on bioinformatics GCB 2007

P-116 Witold Abramowicz, Leszek Maciszek (Eds.) Business Process and Services Computing 1st International Working Conference on Business Process and Services Computing BPSC 2007

P-117 Ryszard Kowalczyk (Ed.) Grid service engineering and manegement The 4th International Conference on Grid Service Engineering and Management GSEM 2007

P-118 Andreas Hein, Wilfried Thoben, Hans-Jürgen Appelrath, Peter Jensch (Eds.) European Conference on ehealth 2007

P-119 Manfred Reichert, Stefan Strecker, Klaus Turowski (Eds.) Enterprise Modelling and Information Systems Architectures Concepts and Applications

P-120 Adam Pawlak, Kurt Sandkuhl, Wojciech Cholewa, Leandro Soares Indrusiak (Eds.) Coordination of Collaborative Engineering - State of the Art and Future Challenges

P-121 Korbinian Herrmann, Bernd Bruegge (Hrsg.) Software Engineering 2008 Fachtagung des GI-Fachbereichs Softwaretechnik

P-122 Walid Maalej, Bernd Bruegge (Hrsg.) Software Engineering 2008 - Workshopband Fachtagung des GI-Fachbereichs Softwaretechnik

P-123 Michael H. Breitner, Martin Breunig, Elgar Fleisch, Ley Pousttchi, Klaus Turowski (Hrsg.)
Mobile und Ubiquitäre Informationssysteme – Technologien, Prozesse, Marktfähigkeit
Proceedings zur 3. Konferenz Mobile und Ubiquitäre Informationssysteme (MMS 2008)

P-124 Wolfgang E. Nagel, Rolf Hoffmann, Andreas Koch (Eds.)
9th Workshop on Parallel Systems and Algorithms (PASA)
Workshop of the GI/ITG Speciel Interest Groups PARS and PARVA

P-125 Rolf A.E. Müller, Hans-H. Sundermeier, Ludwig Theuvsen, Stephanie Schütze, Marlies Morgenstern (Hrsg.)
Unternehmens-IT:
Führungsinstrument oder Verwaltungsbürde
Referate der 28. GIL Jahrestagung

P-126 Rainer Gimnich, Uwe Kaiser, Jochen Quante, Andreas Winter (Hrsg.)
10th Workshop Software Reengineering (WSR 2008)

P-127 Thomas Kühne, Wolfgang Reisig, Friedrich Steimann (Hrsg.)
Modellierung 2008

P-128 Ammar Alkassar, Jörg Siekmann (Hrsg.)
Sicherheit 2008
Sicherheit, Schutz und Zuverlässigkeit
Beiträge der 4. Jahrestagung des Fachbereichs Sicherheit der Gesellschaft für Informatik e.V. (GI)
2.-4. April 2008
Saarbrücken, Germany

P-129 Wolfgang Hesse, Andreas Oberweis (Eds.)
Sigsand-Europe 2008
Proceedings of the Third AIS SIGSAND European Symposium on Analysis, Design, Use and Societal Impact of Information Systems

P-130 Paul Müller, Bernhard Neumair, Gabi Dreo Rodosek (Hrsg.)
1. DFN-Forum Kommunikations-technologien Beiträge der Fachtagung

P-131 Robert Krimmer, Rüdiger Grimm (Eds.)
3rd International Conference on Electronic Voting 2008
Co-organized by Council of Europe, Gesellschaft für Informatik and E-Voting. CC

P-132 Silke Seehusen, Ulrike Lucke, Stefan Fischer (Hrsg.)
DeLFI 2008:
Die 6. e-Learning Fachtagung Informatik

P-133 Heinz-Gerd Hegering, Axel Lehmann, Hans Jürgen Ohlbach, Christian Scheideler (Hrsg.)
INFORMATIK 2008
Beherrschbare Systeme – dank Informatik
Band 1

P-134 Heinz-Gerd Hegering, Axel Lehmann, Hans Jürgen Ohlbach, Christian Scheideler (Hrsg.)
INFORMATIK 2008
Beherrschbare Systeme – dank Informatik
Band 2

P-135 Torsten Brinda, Michael Fothe, Peter Hubwieser, Kirsten Schlüter (Hrsg.)
Didaktik der Informatik –
Aktuelle Forschungsergebnisse

P-136 Andreas Beyer, Michael Schroeder (Eds.)
German Conference on Bioinformatics GCB 2008

P-137 Arslan Brömme, Christoph Busch, Detlef Hühnlein (Eds.)
BIOSIG 2008: Biometrics and Electronic Signatures

P-138 Barbara Dinter, Robert Winter, Peter Chamoni, Norbert Gronau, Klaus Turowski (Hrsg.)
Synergien durch Integration und Informationslogistik
Proceedings zur DW2008

P-139 Georg Herzwurm, Martin Mikusz (Hrsg.)
Industrialisierung des Software-Managements
Fachtagung des GI-Fachausschusses Management der Anwendungsentwick-lung und -wartung im Fachbereich Wirtschaftsinformatik

P-140 Oliver Göbel, Sandra Frings, Detlef Günther, Jens Nedon, Dirk Schadt (Eds.)
IMF 2008 - IT Incident Management & IT Forensics

P-141 Peter Loos, Markus Nüttgens, Klaus Turowski, Dirk Werth (Hrsg.)
Modellierung betrieblicher Informations-systeme (MobIS 2008)
Modellierung zwischen SOA und Compliance Management

P-142 R. Bill, P. Korduan, L. Theuvsen, M. Morgenstern (Hrsg.)
Anforderungen an die Agrarinformatik durch Globalisierung und Klimaveränderung

P-143 Peter Liggesmeyer, Gregor Engels, Jürgen Münch, Jörg Dörr, Norman Riegel (Hrsg.)
Software Engineering 2009
Fachtagung des GI-Fachbereichs Softwaretechnik

P-144 Johann-Christoph Freytag, Thomas Ruf, Wolfgang Lehner, Gottfried Vossen (Hrsg.)
Datenbanksysteme in Business, Technologie und Web (BTW)

P-145 Knut Hinkelmann, Holger Wache (Eds.)
WM2009: 5th Conference on Professional Knowledge Management

P-146 Markus Bick, Martin Breunig, Hagen Höpfner (Hrsg.)
Mobile und Ubiquitäre Informationssysteme – Entwicklung, Implementierung und Anwendung
4. Konferenz Mobile und Ubiquitäre Informationssysteme (MMS 2009)

P-147 Witold Abramowicz, Leszek Maciaszek, Ryszard Kowalczyk, Andreas Speck (Eds.)
Business Process, Services Computing and Intelligent Service Management
BPSC 2009 · ISM 2009 · YRW-MBP 2009

P-148 Christian Erfurth, Gerald Eichler, Volkmar Schau (Eds.)
9th International Conference on Innovative Internet Community Systems
I²CS 2009

P-149 Paul Müller, Bernhard Neumair, Gabi Dreo Rodosek (Hrsg.)
2. DFN-Forum Kommunikationstechnologien
Beiträge der Fachtagung

P-150 Jürgen Münch, Peter Liggesmeyer (Hrsg.)
Software Engineering
2009 - Workshopband

P-151 Armin Heinzl, Peter Dadam, Stefan Kirn, Peter Lockemann (Eds.)
PRIMIUM
Process Innovation for Enterprise Software

P-152 Jan Mendling, Stefanie Rinderle-Ma, Werner Esswein (Eds.)
Enterprise Modelling and Information Systems Architectures
Proceedings of the 3rd Int'l Workshop EMISA 2009

P-153 Andreas Schwill, Nicolas Apostolopoulos (Hrsg.)
Lernen im Digitalen Zeitalter
DeLFI 2009 – Die 7. E-Learning Fachtagung Informatik

P-154 Stefan Fischer, Erik Maehle Rüdiger Reischuk (Hrsg.)
INFORMATIK 2009
Im Focus das Leben

P-155 Arslan Brömme, Christoph Busch, Detlef Hühnlein (Eds.)
BIOSIG 2009:
Biometrics and Electronic Signatures
Proceedings of the Special Interest Group on Biometrics and Electronic Signatures

P-156 Bernhard Koerber (Hrsg.)
Zukunft braucht Herkunft
25 Jahre »INFOS – Informatik und Schule«

P-157 Ivo Grosse, Steffen Neumann, Stefan Posch, Falk Schreiber, Peter Stadler (Eds.)
German Conference on Bioinformatics 2009

P-158 W. Claupein, L. Theuvsen, A. Kämpf, M. Morgenstern (Hrsg.)
Precision Agriculture Reloaded – Informationsgestützte Landwirtschaft

P-159 Gregor Engels, Markus Luckey, Wilhelm Schäfer (Hrsg.)
Software Engineering 2010

P-160 Gregor Engels, Markus Luckey, Alexander Pretschner, Ralf Reussner (Hrsg.)
Software Engineering 2010 – Workshopband
(inkl. Doktorandensymposium)

P-161 Gregor Engels, Dimitris Karagiannis Heinrich C. Mayr (Hrsg.)
Modellierung 2010

P-162 Maria A. Wimmer, Uwe Brinkhoff, Siegfried Kaiser, Dagmar Lück-Schneider, Erich Schweighofer, Andreas Wiebe (Hrsg.)
Vernetzte IT für einen effektiven Staat
Gemeinsame Fachtagung Verwaltungsinformatik (FTVI) und Fachtagung Rechtsinformatik (FTRI) 2010

P-163 Markus Bick, Stefan Eulgem, Elgar Fleisch, J. Felix Hampe, Birgitta König-Ries, Franz Lehner, Key Pousttchi, Kai Rannenberg (Hrsg.)
Mobile und Ubiquitäre Informationssysteme
Technologien, Anwendungen und Dienste zur Unterstützung von mobiler Kollaboration

P-164 Arslan Brömme, Christoph Busch (Eds.)
BIOSIG 2010: Biometrics and Electronic Signatures Proceedings of the Special Interest Group on Biometrics and Electronic Signatures

P-165 Gerald Eichler, Peter Kropf, Ulrike Lechner, Phayung Meesad, Herwig Unger (Eds.)
10th International Conference on Innovative Internet Community Systems (I^2CS) – Jubilee Edition 2010 –

P-166 Paul Müller, Bernhard Neumair, Gabi Dreo Rodosek (Hrsg.)
3. DFN-Forum Kommunikationstechnologien
Beiträge der Fachtagung

P-167 Robert Krimmer, Rüdiger Grimm (Eds.)
4th International Conference on Electronic Voting 2010
co-organized by the Council of Europe, Gesellschaft für Informatik and E-Voting.CC

P-168 Ira Diethelm, Christina Dörge, Claudia Hildebrandt, Carsten Schulte (Hrsg.)
Didaktik der Informatik
Möglichkeiten empirischer Forschungsmethoden und Perspektiven der Fachdidaktik

P-169 Michael Kerres, Nadine Ojstersek Ulrik Schroeder, Ulrich Hoppe (Hrsg.)
DeLFI 2010 - 8. Tagung der Fachgruppe E-Learning der Gesellschaft für Informatik e.V.

P-170 Felix C. Freiling (Hrsg.)
Sicherheit 2010
Sicherheit, Schutz und Zuverlässigkeit

P-171 Werner Esswein, Klaus Turowski, Martin Juhrisch (Hrsg.)
Modellierung betrieblicher Informationssysteme (MobIS 2010)
Modellgestütztes Management

P-172 Stefan Klink, Agnes Koschmider Marco Mevius, Andreas Oberweis (Hrsg.)
EMISA 2010
Einflussfaktoren auf die Entwicklung flexibler, integrierter Informationssysteme
Beiträge des Workshops der GI-Fachgruppe EMISA (Entwicklungsmethoden für Informationssysteme und deren Anwendung)

P-173 Dietmar Schomburg, Andreas Grote (Eds.)
German Conference on Bioinformatics 2010

P-174 Arslan Brömme, Torsten Eymann, Detlef Hühnlein, Heiko Roßnagel, Paul Schmücker (Hrsg.)
perspeGKtive 2010
Workshop „Innovative und sichere Informationstechnologie für das Gesundheitswesen von morgen“

P-175 Klaus-Peter Fähnrich, Bogdan Franczyk (Hrsg.)
INFORMATIK 2010
Service Science – Neue Perspektiven für die Informatik
Band 1

P-176 Klaus-Peter Fähnrich, Bogdan Franczyk (Hrsg.)
INFORMATIK 2010
Service Science – Neue Perspektiven für die Informatik
Band 2

P-177 Witold Abramowicz, Rainer Alt, Klaus-Peter Fähnrich, Bogdan Franczyk, Leszek A. Maciaszek (Eds.)
INFORMATIK 2010
Business Process and Service Science – Proceedings of ISSS and BPSC

P-178 Wolfram Pietsch, Benedikt Krams (Hrsg.)
Vom Projekt zum Produkt
Fachtagung des GI-Fachausschusses Management der Anwendungsentwicklung und -wartung im Fachbereich Wirtschafts-informatik (WI-MAW), Aachen, 2010

P-179 Stefan Gruner, Bernhard Rumpe (Eds.)
FM+AM`2010
Second International Workshop on Formal Methods and Agile Methods

P-180 Theo Härder, Wolfgang Lehner, Bernhard Mitschang, Harald Schöning, Holger Schwarz (Hrsg.)
Datenbanksysteme für Business, Technologie und Web (BTW)
14. Fachtagung des GI-Fachbereichs „Datenbanken und Informationssysteme“ (DBIS)

P-181 Michael Clasen, Otto Schätzel, Brigitte Theuvsen (Hrsg.)
Qualität und Effizienz durch informationsgestützte Landwirtschaft, Fokus: Moderne Weinwirtschaft

P-182 Ronald Maier (Hrsg.)
6th Conference on Professional Knowledge Management
From Knowledge to Action

P-183 Ralf Reussner, Matthias Grund, Andreas Oberweis, Walter Tichy (Hrsg.)
Software Engineering 2011
Fachtagung des GI-Fachbereichs Softwaretechnik

P-184 Ralf Reussner, Alexander Pretschner, Stefan Jähnichen (Hrsg.)
Software Engineering 2011
Workshopband
(inkl. Doktorandensymposium)

P-185 Hagen Höpfner, Günther Specht, Thomas Ritz, Christian Bunse (Hrsg.)
MMS 2011: Mobile und ubiquitäre Informationssysteme Proceedings zur 6. Konferenz Mobile und Ubiquitäre Informationssysteme (MMS 2011)

P-186 Gerald Eichler, Axel Küpper, Volkmar Schau, Hacène Fouchal, Herwig Unger (Eds.)
11th International Conference on Innovative Internet Community Systems (I^2CS)

P-187 Paul Müller, Bernhard Neumair, Gabi Dreo Rodosek (Hrsg.)
4. DFN-Forum Kommunikations-technologien, Beiträge der Fachtagung 20. Juni bis 21. Juni 2011 Bonn

P-188 Holger Rohland, Andrea Kienle, Steffen Friedrich (Hrsg.)
DeLFI 2011 – Die 9. e-Learning Fachtagung Informatik der Gesellschaft für Informatik e.V. 5.–8. September 2011, Dresden

P-189 Thomas, Marco (Hrsg.)
Informatik in Bildung und Beruf INFOS 2011
14. GI-Fachtagung Informatik und Schule

P-190 Markus Nüttgens, Oliver Thomas, Barbara Weber (Eds.)
Enterprise Modelling and Information Systems Architectures (EMISA 2011)

P-191 Arslan Brömme, Christoph Busch (Eds.)
BIOSIG 2011
International Conference of the Biometrics Special Interest Group

P-192 Hans-Ulrich Heiß, Peter Pepper, Holger Schlingloff, Jörg Schneider (Hrsg.)
INFORMATIK 2011
Informatik schafft Communities

P-193 Wolfgang Lehner, Gunther Piller (Hrsg.)
IMDM 2011

P-194 M. Clasen, G. Fröhlich, H. Bernhardt, K. Hildebrand, B. Theuvsen (Hrsg.)
Informationstechnologie für eine nachhaltige Landbewirtschaftung
Fokus Forstwirtschaft

P-195 Neeraj Suri, Michael Waidner (Hrsg.)
Sicherheit 2012
Sicherheit, Schutz und Zuverlässigkeit
Beiträge der 6. Jahrestagung des Fachbereichs Sicherheit der Gesellschaft für Informatik e.V. (GI)

P-196 Arslan Brömme, Christoph Busch (Eds.)
BIOSIG 2012
Proceedings of the 11th International Conference of the Biometrics Special Interest Group

P-197 Jörn von Lucke, Christian P. Geiger, Siegfried Kaiser, Erich Schweighofer, Maria A. Wimmer (Hrsg.)
Auf dem Weg zu einer offenen, smarten und vernetzten Verwaltungskultur
Gemeinsame Fachtagung Verwaltungsinformatik (FTVI) und Fachtagung Rechtsinformatik (FTRI) 2012

P-198 Stefan Jähnichen, Axel Küpper, Sahin Albayrak (Hrsg.)
Software Engineering 2012
Fachtagung des GI-Fachbereichs Softwaretechnik

P-199 Stefan Jähnichen, Bernhard Rumpe, Holger Schlingloff (Hrsg.)
Software Engineering 2012
Workshopband

P-200 Gero Mühl, Jan Richling, Andreas Herkersdorf (Hrsg.)
ARCS 2012 Workshops

P-201 Elmar J. Sinz Andy Schürr (Hrsg.)
Modellierung 2012

P-202 Andrea Back, Markus Bick, Martin Breunig, Key Pousttchi, Frédéric Thiesse (Hrsg.)
MMS 2012:Mobile und Ubiquitäre Informationssysteme

P-203 Paul Müller, Bernhard Neumair, Helmut Reiser, Gabi Dreo Rodosek (Hrsg.)
5. DFN-Forum Kommunikations-technologien
Beiträge der Fachtagung

P-204 Gerald Eichler, Leendert W. M. Wienhofen, Anders Kofod-Petersen, Herwig Unger (Eds.)
12th International Conference on Innovative Internet Community Systems (I2CS 2012)

P-205 Manuel J. Kripp, Melanie Volkamer, Rüdiger Grimm (Eds.)
5th International Conference on Electronic Voting 2012 (EVOTE2012)
Co-organized by the Council of Europe, Gesellschaft für Informatik and E-Voting.CC

P-206 Stefanie Rinderle-Ma, Mathias Weske (Hrsg.)
EMISA 2012
Der Mensch im Zentrum der Modellierung

P-207 Jörg Desel, Jörg M. Haake, Christian Spannagel (Hrsg.)
DeLFI 2012: Die 10. e-Learning Fachtagung Informatik der Gesellschaft für Informatik e.V.
24.–26. September 2012

P-208 Ursula Goltz, Marcus Magnor, Hans-Jürgen Appelrath, Herbert Matthies, Wolf-Tilo Balke, Lars Wolf (Hrsg.)
INFORMATIK 2012

P-209 Hans Brandt-Pook, André Fleer, Thorsten Spitta, Malte Wattenberg (Hrsg.)
Nachhaltiges Software Management

P-210 Erhard Plödereder, Peter Dencker, Herbert Klenk, Hubert B. Keller, Silke Spitzer (Hrsg.)
Automotive – Safety & Security 2012
Sicherheit und Zuverlässigkeit für automobile Informationstechnik

P-211 M. Clasen, K. C. Kersebaum, A. Meyer-Aurich, B. Theuvsen (Hrsg.)
Massendatenmanagement in der Agrar- und Ernährungswirtschaft
Erhebung - Verarbeitung - Nutzung
Referate der 33. GIL-Jahrestagung
20. – 21. Februar 2013, Potsdam

P-212 Arslan Brömme, Christoph Busch (Eds.)
BIOSIG 2013
Proceedings of the 12th International Conference of the Biometrics Special Interest Group
04.–06. September 2013
Darmstadt, Germany

P-213 Stefan Kowalewski, Bernhard Rumpe (Hrsg.)
Software Engineering 2013
Fachtagung des GI-Fachbereichs Softwaretechnik

P-214 Volker Markl, Gunter Saake, Kai-Uwe Sattler, Gregor Hackenbroich, Bernhard Mit schang, Theo Härder, Veit Köppen (Hrsg.)
Datenbanksysteme für Business, Technologie und Web (BTW) 2013
13. – 15. März 2013, Magdeburg

P-215 Stefan Wagner, Horst Lichter (Hrsg.)
Software Engineering 2013
Workshopband
(inkl. Doktorandensymposium)
26. Februar – 1. März 2013, Aachen

P-216 Gunter Saake, Andreas Henrich, Wolfgang Lehner, Thomas Neumann, Veit Köppen (Hrsg.)
Datenbanksysteme für Business, Technologie und Web (BTW) 2013 – Workshopband
11. – 12. März 2013, Magdeburg

P-217 Paul Müller, Bernhard Neumair, Helmut Reiser, Gabi Dreo Rodosek (Hrsg.)
6. DFN-Forum Kommunikations-technologien
Beiträge der Fachtagung
03.–04. Juni 2013, Erlangen

P-218 Andreas Breiter, Christoph Rensing (Hrsg.)
DeLFI 2013: Die 11 e-Learning Fachtagung Informatik der Gesellschaft für Informatik e.V. (GI)
8. – 11. September 2013, Bremen

P-219 Norbert Breier, Peer Stechert, Thomas Wilke (Hrsg.)
Informatik erweitert Horizonte
INFOS 2013
15. GI-Fachtagung Informatik und Schule
26. – 28. September 2013

P-220 Matthias Horbach (Hrsg.)
INFORMATIK 2013
Informatik angepasst an Mensch, Organisation und Umwelt
16. – 20. September 2013, Koblenz

P-221 Maria A. Wimmer, Marijn Janssen, Ann Macintosh, Hans Jochen Scholl, Efthimios Tambouris (Eds.)
Electronic Government and Electronic Participation
Joint Proceedings of Ongoing Research of IFIP EGOV and IFIP ePart 2013
16. – 19. September 2013, Koblenz

P-222 Reinhard Jung, Manfred Reichert (Eds.)
Enterprise Modelling and Information Systems Architectures (EMISA 2013)
St. Gallen, Switzerland
September 5. – 6. 2013

P-223 Detlef Hühnlein, Heiko Roßnagel (Hrsg.)
Open Identity Summit 2013
10. – 11. September 2013
Kloster Banz, Germany

P-224 Eckhart Hanser, Martin Mikusz, Masud Fazal-Baqaie (Hrsg.)
Vorgehensmodelle 2013
Vorgehensmodelle – Anspruch und Wirklichkeit
20. Tagung der Fachgruppe Vorgehensmodelle im Fachgebiet Wirtschaftsinformatik (WI-VM) der Gesellschaft für Informatik e.V.
Lörrach, 2013

P-225 Hans-Georg Fill, Dimitris Karagiannis, Ulrich Reimer (Hrsg.)
Modellierung 2014
19. – 21. März 2014, Wien

P-226 M. Clasen, M. Hamer, S. Lehnert, B. Petersen, B. Theuvsen (Hrsg.)
IT-Standards in der Agrar- und Ernährungswirtschaft Fokus: Risiko- und Krisenmanagement
Referate der 34. GIL-Jahrestagung
24. – 25. Februar 2014, Bonn

P-227 Wilhelm Hasselbring,
Nils Christian Ehmke (Hrsg.)
Software Engineering 2014
Fachtagung des GI-Fachbereichs Softwaretechnik
25. – 28. Februar 2014
Kiel, Deutschland

P-228 Stefan Katzenbeisser, Volkmar Lotz, Edgar Weippl (Hrsg.)
Sicherheit 2014
Sicherheit, Schutz und Zuverlässigkeit
Beiträge der 7. Jahrestagung des Fachbereichs Sicherheit der Gesellschaft für Informatik e.V. (GI)
19. – 21. März 2014, Wien

P-229 Dagmar Lück-Schneider, Thomas Gordon, Siegfried Kaiser, Jörn von Lucke,Erich Schweighofer, Maria A.Wimmer, Martin G. Löhe (Hrsg.)
Gemeinsam Electronic Government ziel(gruppen)gerecht gestalten und organisieren
Gemeinsame Fachtagung Verwaltungsinformatik (FTVI) und Fachtagung Rechtsinformatik (FTRI) 2014, 20.-21. März 2014 in Berlin

P-230 Arslan Brömme, Christoph Busch (Eds.)
BIOSIG 2014
Proceedings of the 13th International Conference of the Biometrics Special Interest Group
10. – 12. September 2014 in Darmstadt, Germany

P-231 Paul Müller, Bernhard Neumair, Helmut Reiser, Gabi Dreo Rodosek (Hrsg.)
7. DFN-Forum Kommunikationstechnologien
16. – 17. Juni 2014
Fulda

P-232 E. Plödereder, L. Grunske, E. Schneider, D. Ull (Hrsg.)
INFORMATIK 2014
Big Data – Komplexität meistern
22. – 26. September 2014
Stuttgart

P-233 Stephan Trahasch, Rolf Plötzner, Gerhard Schneider, Claudia Gayer, Daniel Sassiat, Nicole Wöhrle (Hrsg.)
DeLFI 2014 – Die 12. e-Learning Fachtagung Informatik
der Gesellschaft für Informatik e.V.
15. – 17. September 2014
Freiburg

P-234 Fernand Feltz, Bela Mutschler, Benoît Otjacques (Eds.)
Enterprise Modelling and Information Systems Architectures
(EMISA 2014)
Luxembourg, September 25-26, 2014

P-235 Robert Giegerich, Ralf Hofestädt, Tim W. Nattkemper (Eds.)
German Conference on Bioinformatics 2014
September 28 – October 1
Bielefeld, Germany

P-236 Martin Engstler, Eckhart Hanser, Martin Mikusz, Georg Herzwurm (Hrsg.)
Projektmanagement und Vorgehensmodelle 2014
Soziale Aspekte und Standardisierung
Gemeinsame Tagung der Fachgruppen Projektmanagement (WI-PM) und Vorgehensmodelle (WI-VM) im Fachgebiet Wirtschaftsinformatik der Gesellschaft für Informatik e.V., Stuttgart 2014

P-237 Detlef Hühnlein, Heiko Roßnagel (Hrsg.)
Open Identity Summit 2014
4.–6. November 2014
Stuttgart, Germany

P-238 Arno Ruckelshausen, Hans-Peter Schwarz, Brigitte Theuvsen (Hrsg.)
Informatik in der Land-, Forst- und Ernährungswirtschaft
Referate der 35. GIL-Jahrestagung
23. – 24. Februar 2015, Geisenheim

P-239 Uwe Aßmann, Birgit Demuth, Thorsten Spitta, Georg Püschel, Ronny Kaiser (Hrsg.)
Software Engineering & Management 2015
17.-20. März 2015, Dresden

P-240 Herbert Klenk, Hubert B. Keller, Erhard Plödereder, Peter Dencker (Hrsg.)
Automotive – Safety & Security 2015
Sicherheit und Zuverlässigkeit für automobile Informationstechnik
21.–22. April 2015, Stuttgart

P-241 Thomas Seidl, Norbert Ritter, Harald Schöning, Kai-Uwe Sattler, Theo Härder, Steffen Friedrich, Wolfram Wingerath (Hrsg.)
Datenbanksysteme für Business, Technologie und Web (BTW 2015)
04. – 06. März 2015, Hamburg

P-242 Norbert Ritter, Andreas Henrich, Wolfgang Lehner, Andreas Thor, Steffen Friedrich, Wolfram Wingerath (Hrsg.)
Datenbanksysteme für Business, Technologie und Web (BTW 2015) – Workshopband
02. – 03. März 2015, Hamburg

P-243 Paul Müller, Bernhard Neumair, Helmut Reiser, Gabi Dreo Rodosek (Hrsg.)
8. DFN-Forum Kommunikationstechnologien
06.–09. Juni 2015, Lübeck

P-244 Alfred Zimmermann, Alexander Rossmann (Eds.)
Digital Enterprise Computing (DEC 2015)
Böblingen, Germany June 25-26, 2015

P-245 Arslan Brömme, Christoph Busch , Christian Rathgeb, Andreas Uhl (Eds.)
BIOSIG 2015
Proceedings of the 14th International Conference of the Biometrics Special Interest Group
09.–11. September 2015
Darmstadt, Germany

P-246 Douglas W. Cunningham, Petra Hofstedt, Klaus Meer, Ingo Schmitt (Hrsg.)
INFORMATIK 2015
28.9.-2.10. 2015, Cottbus

P-247 Hans Pongratz, Reinhard Keil (Hrsg.)
DeLFI 2015 – Die 13. E-Learning Fachtagung Informatik der Gesellschaft für Informatik e.V. (GI)
1.–4. September 2015
München

P-248 Jens Kolb, Henrik Leopold, Jan Mendling (Eds.)
Enterprise Modelling and Information Systems Architectures
Proceedings of the 6th Int. Workshop on Enterprise Modelling and Information Systems Architectures, Innsbruck, Austria
September 3-4, 2015

P-249 Jens Gallenbacher (Hrsg.)
Informatik allgemeinbildend begreifen
INFOS 2015 16. GI-Fachtagung Informatik und Schule
20.–23. September 2015

P-250 Martin Engstler, Masud Fazal-Baqaie, Eckhart Hanser, Martin Mikusz, Alexander Volland (Hrsg.)
Projektmanagement und Vorgehensmodelle 2015
Hybride Projektstrukturen erfolgreich umsetzen
Gemeinsame Tagung der Fachgruppen Projektmanagement (WI-PM) und Vorgehensmodelle (WI-VM) im Fachgebiet Wirtschaftsinformatik der Gesellschaft für Informatik e.V., Elmshorn 2015

P-251 Detlef Hühnlein, Heiko Roßnagel, Raik Kuhlisch, Jan Ziesing (Eds.)
Open Identity Summit 2015
10.–11. November 2015
Berlin, Germany

P-252 Jens Knoop, Uwe Zdun (Hrsg.)
Software Engineering 2016
Fachtagung des GI-Fachbereichs Softwaretechnik
23.–26. Februar 2016, Wien

P-253 A. Ruckelshausen, A. Meyer-Aurich, T. Rath, G. Recke, B. Theuvsen (Hrsg.)
Informatik in der Land-, Forst- und Ernährungswirtschaft
Fokus: Intelligente Systeme – Stand der Technik und neue Möglichkeiten
Referate der 36. GIL-Jahrestagung
22.-23. Februar 2016, Osnabrück

P-254 Andreas Oberweis, Ralf Reussner (Hrsg.)
Modellierung 2016
2.–4. März 2016, Karlsruhe

P-255 Stefanie Betz, Ulrich Reimer (Hrsg.)
Modellierung 2016 Workshopband
2.–4. März 2016, Karlsruhe

P-256 Michael Meier, Delphine Reinhardt, Steffen Wendzel (Hrsg.)
Sicherheit 2016
Sicherheit, Schutz und Zuverlässigkeit
Beiträge der 8. Jahrestagung des Fachbereichs Sicherheit der Gesellschaft für Informatik e.V. (GI)
5.–7. April 2016, Bonn

P-257 Paul Müller, Bernhard Neumair, Helmut Reiser, Gabi Dreo Rodosek (Hrsg.)
9. DFN-Forum Kommunikationstechnologien
31. Mai – 01. Juni 2016, Rostock

P-258 Dieter Hertweck, Christian Decker (Eds.)
Digital Enterprise Computing (DEC 2016)
14.–15. Juni 2016, Böblingen

P-259 Heinrich C. Mayr, Martin Pinzger (Hrsg.)
INFORMATIK 2016
26.–30. September 2016, Klagenfurt

P-260 Arslan Brömme, Christoph Busch, Christian Rathgeb, Andreas Uhl (Eds.)
BIOSIG 2016
Proceedings of the 15th International Conference of the Biometrics Special Interest Group
21.–23. September 2016, Darmstadt

P-261 Detlef Rätz, Michael Breidung, Dagmar Lück-Schneider, Siegfried Kaiser, Erich Schweighofer (Hrsg.)
Digitale Transformation: Methoden, Kompetenzen und Technologien für die Verwaltung
Gemeinsame Fachtagung Verwaltungsinformatik (FTVI) und Fachtagung Rechtsinformatik (FTRI) 2016
22.–23. September 2016, Dresden

P-262 Ulrike Lucke, Andreas Schwill, Raphael Zender (Hrsg.)
DeLFI 2016 – Die 14. E-Learning Fachtagung Informatik
der Gesellschaft für Informatik e.V. (GI)
11.–14. September 2016, Potsdam

P-263 Martin Engstler, Masud Fazal-Baqaie, Eckhart Hanser, Oliver Linssen, Martin Mikusz, Alexander Volland (Hrsg.)
Projektmanagement und Vorgehensmodelle 2016
Arbeiten in hybriden Projekten: Das Sowohl-als-auch von Stabilität und Dynamik
Gemeinsame Tagung der Fachgruppen Projektmanagement (WI-PM) und Vorgehensmodelle (WI-VM) im Fachgebiet Wirtschaftsinformatik
der Gesellschaft für Informatik e.V., Paderborn 2016

P-264 Detlef Hühnlein, Heiko Roßnagel, Christian H. Schunck, Maurizio Talamo (Eds.)
Open Identity Summit 2016
der Gesellschaft für Informatik e.V. (GI)
13.–14. October 2016, Rome, Italy

P-265 Bernhard Mitschang, Daniela Nicklas,Frank Leymann, Harald Schöning, Melanie Herschel, Jens Teubner, Theo Härder, Oliver Kopp, Matthias Wieland (Hrsg.)
Datenbanksysteme für Business, Technologie und Web (BTW 2017)
6.–10. März 2017, Stuttgart

P-266 Bernhard Mitschang, Norbert Ritter, Holger Schwarz, Meike Klettke, Andreas Thor, Oliver Kopp, Matthias Wieland (Hrsg.)
Datenbanksysteme für Business, Technologie und Web (BTW 2017)
Workshopband
6.–7. März 2017, Stuttgart

P-267 Jan Jürjens, Kurt Schneider (Hrsg.)
Software Engineering 2017
21.–24. Februar 2017, Hannover

P-268 A. Ruckelshausen, A. Meyer-Aurich, W. Lentz, B. Theuvsen (Hrsg.)
Informatik in der Land-, Forst- und Ernährungswirtschaft
Fokus: Digitale Transformation – Wege in eine zukunftsfähige Landwirtschaft
Referate der 37. GIL-Jahrestagung
06.–07. März 2017, Dresden

P-269 Peter Dencker, Herbert Klenk, Hubert Keller, Erhard Plödereder (Hrsg.)
Automotive – Safety & Security 2017
30.–31. Mai 2017, Stuttgart

P-270 Arslan Brömme, Christoph Busch, Antitza Dantcheva, Christian Rathgeb, Andreas Uhl (Eds.)
BIOSIG 2017
20.–22. September 2017, Darmstadt

P-271 Paul Müller, Bernhard Neumair, Helmut Reiser, Gabi Dreo Rodosek (Hrsg.)
10. DFN-Forum Kommunikations-technologien
30. – 31. Mai 2017, Berlin

P-272 Alexander Rossmann, Alfred Zimmermann (eds.)
Digital Enterprise Computing (DEC 2017)
11.–12. Juli 2017, Böblingen

P-273 Christoph Igel, Carsten Ullrich, Martin Wessner (Hrsg.)
BILDUNGSRÄUME
DeLFI 2017
Die 15. e-Learning Fachtagung Informatik der Gesellschaft für Informatik e.V. (GI)
5. bis 8. September 2017, Chemnitz

P-274 Ira Diethelm (Hrsg.)
Informatische Bildung zum Verstehen und Gestalten der digitalen Welt
13.–15. September 2017, Oldenburg

P-275 Maximilian Eibl, Martin Gaedke (Hrsg.)
INFORMATIK 2017
25.–29. September 2017, Chemnitz

P276 Alexander Volland, Martin Engstler, Masud Fazal-Baqaie, Eckhart Hanser, Oliver Linssen, Martin Mikusz (Hrsg.)
Projektmanagement und Vorgehensmodelle 2017
Die Spannung zwischen dem Prozess und den Menschen im Projekt
Gemeinsame Tagung der Fachgruppen Projektmanagement und Vorgehensmodelle im Fachgebiet Wirtschaftsinformatik der Gesellschaft für Informatik e.V. in Kooperation mit der Fachgruppe IT-Projektmanagement der GPM e.V., Darmstadt 2017

P-277 Lothar Fritsch, Heiko Roßnagel, Detlef Hühnlein (Hrsg.)
Open Identity Summit 2017
5.–6. October 2017, Karlstad, Sweden

P-278 Arno Ruckelshausen, Andreas Meyer-Aurich, Karsten Borchard, Constanze Hofacker, Jens-Peter Loy, Rolf Schwerdtfeger, Hans-Hennig Sundermeier, Helga Floto, Brigitte Theuvsen (Hrsg.)
Informatik in der Land-, Forst- und Ernährungswirtschaft
Referate der 38. GIL-Jahrestagung
26.–27. Februar 2018, Kiel

P-279 Matthias Tichy, Eric Bodden, Marco Kuhrmann, Stefan Wagner, Jan-Philipp Steghöfer (Hrsg.)
Software Engineering und Software Management 2018
5.–9. März 2018, Ulm

P-280 Ina Schaefer, Dimitris Karagiannis, Andreas Vogelsang, Daniel Méndez, Christoph Seidl (Hrsg.)
Modellierung 2018
21.–23. Februar 2018, Braunschweig

P-281 Hanno Langweg, Michael Meier, Bernhard C. Witt, Delphine Reinhardt (Hrsg.)
Sicherheit 2018
Sicherheit, Schutz und Zuverlässigkeit
25.–27. April 2018, Konstanz

P-283 Paul Müller, Bernhard Neumair, Helmut Reiser, Gabi Dreo Rodosek (Hrsg.)
11. DFN-Forum Kommunikationstechnologien
27.–28. Juni 2018, Günzburg

The titles can be purchased at:

Köllen Druck + Verlag GmbH
Ernst-Robert-Curtius-Str. 14 · D-53117 Bonn
Fax: +49 (0)228/9898222
E-Mail: druckverlag@koellen.de