An Agent-based Framework for a Decentralized Reconstruction of Attack Paths

Mario Golling, Robert Koch, Frank Tietze, Sandy-Dorothea Hein,
Michael Kretzschmar and Gabi Dreo Rodosek

Munich Network Management Team (MNM-Team)
Universität der Bundeswehr München
Werner-Heisenberg-Weg 39
D-85577 Neubiberg
{mario.golling, robert.koch, frank.tietze, sandy-dorothea.hein,
michael.kretzschmar, gabi.dreo}@unibw.de

Abstract: Network forensics is gaining higher importance throughout the last years due to the ever-growing number of attacks. A big driving factor in that process next to what is technically possible is also what is legally feasible. In contrast to conventional network forensics, which relies on a central approach, both legal as well as technical guidelines nowadays favor a decentralized approach since aspects like privacy, limited data manipulation possibilities and scalability are addressed superiorly. Even though some decentralized approaches are already available, especially in the area of protection against manipulation, i.e., falsification of relevant forensics data, all of them are in the need of improvement particularly in case of sophisticated attacks. For this reason, this paper is devoted to decentralized network forensics and proposes its own agent-based framework for an automated reconstruction of attack paths. Using a scenario, requirements are derived on which state of the art approaches are briefly examined and evaluated before our own concept is presented. Thereafter, the proof of concept is illustrated before the work is concluded.

1 Introduction

As a result of the increasingly important role of computer systems in our society - particularly in modern companies - they are also used more and more to commit crimes. Although several technical protective measures such as virus scanners, firewalls, intrusion detection systems (IDSs) or access controls list (ACLs) try to prevent causing crimes like the illegal interception or the theft of sensitive data for industrial espionage, both quantity and quality of attacks are increasing steadily [KSG12]. In order to find the perpetrator and prove the offense, a forensics investigation may be initiated in case of an incident. In the scope of this paper, we herby focus solely on network forensics, *the science that deals with capture, recording, and analysis of network traffic for detecting intrusions and investigating them* [PJN10]. In this regard, various sources of information are/can be used within the forensic process: flow records, IP routing tables, ARP tables, spanning tree information or additional information of dynamic routing protocols / layer 2 redundancy protocols such

as the Virtual Switch Redundancy Protocol (VSRP), to name a view (e.g., see [Sys11] for more information).

Problem Statement
In contrast to conventional network forensics which relies on a central approach (i.e. where data is stored and analyzed centrally), both legal as well as technical experts nowadays favor a decentralized approach since aspects like privacy, limited data manipulation possibilities and scalability are addressed superiorly. In terms of *legal restrictions* within the European Union for instance, the upcoming EU General Data Protection Regulation [Alb13, Alb14], explicitly refers to privacy by design/by default as one of the core elements. Although not explicitly part of the legal text, a vast majority of experts believe that the idea of privacy by design, among others, favors a *distributed approach* [GGTD11]. With regard to the *technical guidelines*, we like to point at [NJM+12] for instance. According to this reference, a multitude of privacy breaches has spurred research into privacy-preserving alternatives for centralized data storing, exploring a number of techniques for storing, disseminating, and controlling access to data in a *distributed fashion*.

Outline of the Paper
The aim of this paper is therefore to carry out decentralized network forensics and in particular to address the aspect of protection against manipulation. To this end, the paper starts with identifying requirements for decentralized network forensics, using a real-world sophisticated attack in Section 2. In Section 3, this paper shortly investigates existing approaches. Thereafter, within Section 4, our own framework is presented. Following this, Section 5 presents the corresponding Proof of Concept (PoC), before the conclusions are drawn in Section 6.

2 Requirements Analysis

Scenario
For a better understanding and to evaluate existing approaches, a small scenario of a sophisticated attack is used in the following (see Figure 1 for a graphical overview).

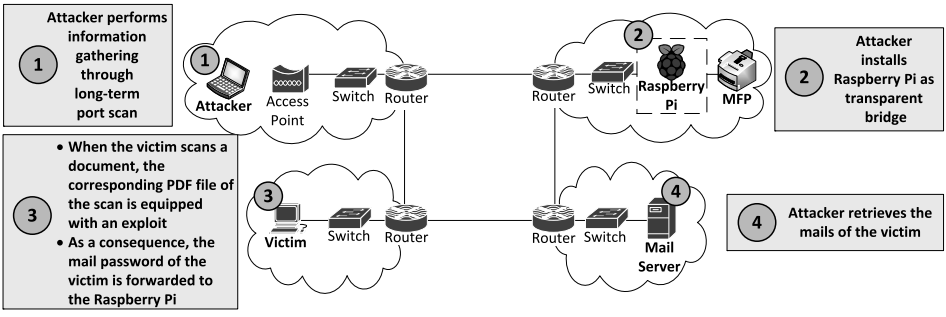

Figure 1: Scenario of a Sophisticated Attack

Within the first step, the attacker performs port scans (unnoticed due to the fact that they are performed in a time frame of multiple weeks) in order to come to a more accurate picture of the network infrastructure of his university and to complete publicly available network information. Based on this, the attacker installs a Raspberry Pi (shorty referred to as PI within the rest of this paper) as a transparent bridge between a multi-function printer (MFP with scan-to-mail) and the switch (resp. the LAN socket) with the result that all data communications are monitored. In addition to passive monitoring, the attacker is also able to change the traffic if needed. To this end, he infects a PDF file with malicious code while it is sent within the scan-to-email function of the printer by the victim (who is previously specifically chosen by the attacker). By this malicious code, an exploit is executed on the victim's computer, sending the mail password of the victim to the PI. Therefore, the PI, once a connection is initiated to a pre-selected TCP Port, does not behave as a bridge, but also actively responds to connection requests). After having removed the PI, the attacker then fetches the e-mails of the victim to gain more important information about an upcoming exam.
Reflecting the scenario in particular the following points are essential. Data which is essential for the creation of a complete picture of the situation is available (i) in different places (location), (ii) in different types (Flow data, Spanning Tree, log files, etc.) (iii) at various levels of detail (different log levels) (iv) with different duration (volatile vs. non-volatile; different extinguishing periods). Furthermore, it is striking that the process of creating a complete picture of the situation, even at this relatively simple scenario, is by no means trivial, and therefore (i) the analyst has to be supported (complexity reduction) and (ii) already a few measures of the attacker create a high level of distortion which in turn demands that previous analysis results have to be verified / falsified.

Requirements
These considerations are addressed by the following points: the general requirement of "decentralization" can be further subdivided into three other criteria, which are, on the one hand, very important to evaluate existing works in a better way and on the other hand, clarifies the underlying importance.

- *Distributed Storage* refers to the idea that data should *not* be stored on a central server (a syslog server for example), but instead remain on multiple systems. Referring to the scenario, the data should be stored locally on the switches, routers, etc., and only (i) "merged" when necessary and only (ii) the pieces of data that are really needed for the analysis (case by case) should be sent.
- *Distributed Data Processing:* closely related to the concept of distributed storage is the concept of distributed analysis. This means that processing such as filtering of personal data has to be decentralized, too (and should therefore be done on the remote devices as well). Processing is not limited to extracting useful data from raw network data, but also implies processing a request and sending only evidence data to the centralized data storage when needed.
- *Protection Against Manipulation:* once data is stored in a decentralized fashion, an attacker has to overcome several hurdles to cover his tracks. Unfortunately, it may also happen in a decentralized world that individual elements (routers, switches, firewalls, etc.) are victims of an attack. Protection against manipulation is therefore

an important requirement for the determination of the trustworthiness of the involved network components as data may be falsified (e.g., as a result of a compromise of a device). Thus, the data received from the (potentially) malicious device has to verified/falsified. This is particularly relevant if not all information is available (e.g., because data has been overwritten in the meantime). In addition, all data is supposed to be read or modified solely by authorized users.

Next to this, additional criteria also have to be considered; e.g., see [GS11, KGH11]:

- *Search Function* expresses whether the given architecture has sophisticated search capabilities. When an incident occurs, a search request should at least be based on source IP/port, destination IP/port and time slot.
- *Automated Attack Path Reconstruction*: attribution of an attack is the main task of IT forensics. To identify the actual actor, the complete attack path, thus the way the packet has taken, has to be reconstructed backwards from the target through various intermediate components (such as routers, switches, firewalls, gateways, etc.) to the attacker.
- *Documentation/Presentation:* the involved network components have to log every forensics activity such as accesses, modifications and requests together with a time-stamp and the activity itself. In addition, the forensic investigator has to document the activities and the results in detail. Thus, the investigation can be comprehended and rerun (which is important for an usability as evidence in court). Next to this, the analyst has to prepare presentations of the results for different target groups. Both documentation and presentation shall leave as little as possible room for interpretation. Although this is primarily a requirement for the forensic investigator, the underlying forensics software has to support this.

3 Related Work

Due to space limitations, only a short excerpt of related work in the field of decentralized network forensics can be presented here. For a more comprehensive overview please take a look at, e.g., [PJN10].

As particularly illustrated by Table 1, decentralized approaches can be classified into (i) those that store resp. (ii) evaluate data in a distributed fashion, or (iii) those that perform operations distributedly. For and on behalf of the first six approaches we like to point out the approach of Ren in more detail. In 2004, Ren has published his reference model of a *distributed cooperative network forensics system* based on a client-server architecture [Ren04]. Here, clients are distributed agents that integrate data from IDSs, firewalls, honeynets and remote traffic. The central server captures, filters, dumps and transforms the network traffic and analyzes the extracted and transformed data to replay the network behavior [PJN10]. From the second six approaches, i.e. those that are more decentralized, we would like to point out Tang and Daniels' approach in more detail. In Tang and Daniels' *simple decentralized framework for network forensics* [TD05] agents are assigned to single sub-networks where they monitor the network traffic and store the data locally. Multiple agents are under control of a proxy and send the data (based on filter criteria like systems/networks).

Table 1: Evaluation of Distributed State of the Art Approaches

Approach	Distributed Storage	Distributed Data Processing	Protection Against Manipulation	Search Function	Automated Attack Path Reconstruction	Documentation/ Presentation
Ren	-	-	-	(√)	√	√
Wang et al.	-	-	-	√	√	(√)
Lin et al.	-	-	-	(√)	√	(√)
Wang and Li	-	-	-	(√)	-	(√)
Masti et al.	-	-	-	(√)	√	(√)
Hong et al.	-	-	-	√	√	√
Scanlon and Kechadi	-	-	-	(√)	-	√
Hoelz et al.	-	√	-	√	√	(√)
Shanmugasundaram et al.	√	√	-	√	-	(√)
Ren and Jin	√	√	(√)	√	√	√
Tang and Daniels	√	√	-	√	√	(√)
Nagesh	√	√	-	(√)	-	(√)
Armoogum and Mohamudally	√	√	-	(√)	√	(√)

As also depicted in Table 1, current solutions don't provide sophisticated built-in capabilities for a (semi-)automated protection against manipulation. As it can not be excluded that some systems are compromised and thus as a consequence send incorrect data for further analysis, this aspect should be given more importance.

4 Concept

To close the gaps shown in the last section, our concept is primarily based on four considerations: (i) a systematic decentralization and a network concept with consistent security by design, in which security consequently moves in the center of attention right from the beginning, (ii) a systematic questioning of analysis results in relation to their plausibility, (iii) an use of standards as early as possible (and thus the elimination of possibly existing vendor lock-ins) and (iv) a reasonable rights and role concept (which explicitly includes temporary assignments of rights).

4.1 Network Concept

Belonging to the phase of strategic analysis, the network concept has a significant impact on the data that can be used later for analysis purposes, both in terms of quality and quantity [GKHR13]. As depicted in Figure 2, we propose a strict separation between the forensic network and the networks used productively. One core element is the use of PIs, which (i) are representing the link between the network component and the forensics network, (ii) collect the (forensic) data from the corresponding device (i.e. extract and store the data from the respective device) and (iii) perform a decentralized analysis.
For this purpose, the PIs are dual-homed. On the one side, the PIs are connected to the corresponding network component of the productive network via a LAN cable and, if necessary, a RS232 interface (*data recording*). On the other side, the PIs are connected to the forensic network via LAN cable only (*forensics layer*). As Section 5 will illustrate in more detail, we have installed a syslog server on each PI where the data of the corresponding network component is stored locally. If not all relevant information is transmitted via the syslog protocol by the respective network components manufacturer, our approach explicitly

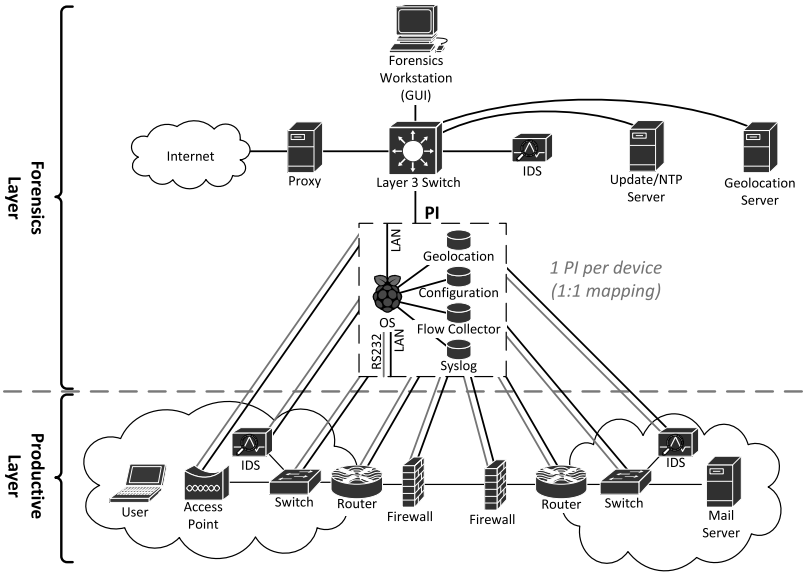

Figure 2: Network Concept

proposes custom actions. For this, either via LAN (e.g., using Telnet or SSH) or via RS232, a connection is established to the component and based on that, the necessary data is extracted from the console (if possible). We also want to overcome any vendor lock-ins as early as possible and thus only make use of proprietary forms of communication between the PI and the associated network component. In this context, we have also - which is also discussed in more detail in Section 5 - installed Web Service Agents on the PIs in order to perform the distributed analysis (in a standardized form/with standardized protocols). In addition to this, flow collectors are also installed on the PIs to store the flow data accordingly. Following the idea of one of our former publications [KGSR] (which introduces the concept of geo-based scoring/reputation, however, in a centralized form), we also propose to make use of this in a decentralized form. For this purpose, analogous to the idea of IP scoring, depending on the geo-based scoring (and thus the suspiciousness of the geographical area), the amount of the data recorded is set; so, the more suspicious the geographical area, the more data is stored. For the sake of completeness, it should be mentioned here that the aspect of privacy is extremely important for us and, although our general approach is to store data from as many sources as possible (*broad data base*), we focus on meta-data as far as possible. Only in exceptional cases (i.e. when a very high degree of suspiciousness is given), the complete data stream is recorded/analyzed. Furthermore, legal erase / blocking respites are also implemented locally.

For distributed data analysis, the PIs are connected to the analysis network with a switch supporting extended ACLs and thus explicitly preventing any communication among the PIs (with the exception of a communication to the PIs of the immediate neighbors; for an explanation see below). Next to this, the PIs make use of the IPsec Authentication

Header Processing (Transport Mode, SHA-256) in order to explicitly address the aspect of authentication/integrity. For the sake of clarity, it should be mentioned that no encryption shall be used (to make intrusion detection more easy).
Furthermore, the forensic network also consists of (i) the forensic workstation (which merges the data from the PIs, implements a federated (temporary) rights and role concept and visualizes the results), (ii) the Geolocation server (which constantly updates the central Geolocation database and makes it available to the PIs), (iii) a special update server (to keep the PIs up to date), (iv) an IDS (to be able to detect attacks within the forensic network) and (v) a proxy server (which in turn ensures access to the Internet for the Geolocation server, the update server and the IDS, as the IDS also has to be updated continuously with new signatures).
To be able to include external investigators within the forensics analysis (which is often needed esp. in smaller firms), a rights and role concept is implemented. This aspect is taken care by using Shibboleth, a state-of-the-art software for the so-called federated identity management [MCC+04]. This allows for cross-domain single sign-on and removes the need to maintain user names and passwords.

4.2 Plausibility Checks

Unfortunately, it may also happen that messages of network elements and/or PIs are incorrect. Possible reasons for this are software/configuration errors, but also attacks on the network infrastructure. In particular, if not all data is available (either because it has been deleted in the meantime or - for whatever reason - was not even collected), a malicious device can create false attack traces, especially in case of a sophisticated attack. To avoid this situation, our architecture autonomously performs plausibility checks to support the administrator in identifying non-trivial errors. Therefore, each analysis result is submitted to the immediate neighbors and is automatically verified/falsified (see Figure 3). Immediate neighbors are defined as all direct neighbors of the corresponding

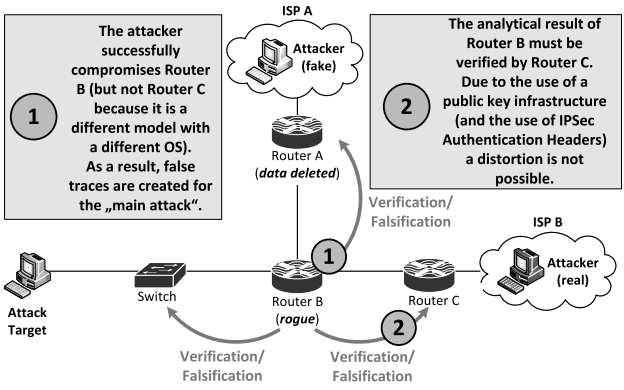

Figure 3: Plausibility Checks

device, even if there was no *active* connection to that device in the evaluation-timeframe

(accordingly, a verification/falsification is performed by making use of the the flooding principle). In addition to this verification/falsification (performed whenever a forensic analysis is performed) the PIs sporadically initiate a review of the trust level of their neighbors (by autonomously starting the process).

5 Prototype

The concept developed in Section 4 was implemented prototypically and is presented as follows.

5.1 Initial Configuration

In the beginning, the PIs are updated and all necessary tools are downloaded: wireshark, minicom, nfdump (for NetFlow) or nfdump-sFlow (for sFlow) and atftp. With minicom, a connection between the PI and the router or switch can be set up via the console port. Furthermore, a flow collector (nfcapd for NetFlow resp. sfcapd for sFlow) is configured on each PI. Afterwards, the syslog daemon is configured. In order to exchange data between the PI and the network device, Advanced TFTP (atftp) is used.

5.2 Implementation of the Web Service

After the successful configuration of the network environment, Web Services are implemented. Here, Eclipse has been chosen as development software and consequently needs to be installed together with the feature "Web, XML, Java EE and OSGi Enterprise Development". Furthermore, OpenJDK (in our PoC: 1.6.0) and Apache Tomcat (in our PoC: 7) are to be installed.

Raspberry Pi

The classes for the Web Service provider have been programmed first. The chosen programming language is Java. A dynamic Web Service project was created and in particular the following classes were implemented:

- *Flow:* this class defines the parameters of a flow.
- *FlowFilter:* this class defines the parameters of a flow filter that can be set along with getters and setters.
- *SearchFlows:* this class searches the flow dumps based on the set flow filter. Therefore a command for nfdump is generated with the help of the filter parameters. The command returns only matching flows that are put in a HashMap which is then returned for further processing.
- *Syslog:* this class defines the parameters of a syslog entry along with getters and setters. For instance, the parameters for a Cisco router are: timestamp, IP address, facility, severity, type of message and message.
- *SyslogFilter:* this class defines the parameters (timestamp under specification of year, month and day) of a syslog filter that can be set along with getters and setters.
- *SearchLogs:* this class reads the log files, compares the set date with the date of the log entries and returns the filtered matches in a HashMap.

- *CreateXML:* this class creates and saves an XML file based on the information is extracted from the respective HashMaps flowMap and logMap.
- *SearchConfigs:* This class reads the configuration file of the respective network device and puts it into a string which is then returned.
- *NetworkDevicesData:* this class is the main class and sets the filter parameters based on the request of the master, initiates the creation of the XML file and returns the configuration of the respective network device.

Forensics Workstation
After generation of the Web Service client, some modifications are made on the Java Server Pages (JSP) files (*excerpt*):

- *TestClient:* this file defines the layout of the client.
- *Method:* the only method that is specified in this file is the one to set the filter parameters.
- *Input:* this file creates the graphic representation for the request.
- *Result:* with this file, the reply to the request is prepared in terms of graphics and content.

5.3 Evaluation

The PoC was evaluated using two test environments. Test environment one included two PIs each with a 16 GB SD card, one computer running Ubuntu as forensics workstation, one Cisco 1841 router plus one HP ProCurve Switch 3400cl. Here, the focus was on recording and transferring of relevant data such as flow exports, IP routing tables, ARP tables and the running configuration. The second test environment included 4 Cisco 1841 Router, 2 HP ProCurve Switches 3400c and 2 HP ProCurve Switches 5400zl and 8 PIs. In this setup, especially the behavior of the architecture in case of compromised devices has been studied extensively (see again Figure 3). Although at this point, unfortunately, due to space limitations, we can not discuss results / test runs in detail, the conceptual idea of plausibility checks has been successfully transferred into practice.

6 Conclusions and Outlook

Successful attacks on IT infrastructures are lately increasing rapidly. With this, the need for forensic solutions is growing. With the upcoming introduction of the new EU Data Protection Directive, the concept of distributed storage and analysis is of high relevance. Current solutions in this area already provide some important features, but, however, fail to address important aspects in terms of protection against manipulation. To fill this gap, a concept and a prototypical implementation was presented within this paper including the first promising extracts from the evaluation. The next majors steps of the implementation will be to improve the graphical representation as well as the stability of the program. In the context of lawful interception, the results of this paper could be transferred as well. Although lawful interception implies organized crime or terrorist activities, it demands similar conceptual requirements.

Acknowledgements

This work was partly funded by FLAMINGO, a Network of Excellence project (ICT-318488) supported by the European Commission under its Seventh Framework Programme.

References

[Alb13] Jan Philipp Albrecht. Report on the Proposal for a Regulation on the Protection of Individuals with Regard to the Processing of Personal Data. Technical report, European Parliament, 2013.

[Alb14] Jan Philipp Albrecht. Report on the Proposal for a Regulation of the European Parliament and of the Council on the Protection of Individuals with Regard to the Processing of Personal Data. Technical report, Council of the European Union, 2014.

[GGTD11] Seda Gürses, Carmela Gonzalez Troncoso, and Claudia Diaz. Engineering Privacy by Design. *Computers, Privacy & Data Protection*, 2011.

[GKHR13] Mario Golling, Robert Koch, Peter Hillmann, and Gabi Dreo Rodosek. Ganzheitliche Architektur zur Entwicklung und Analyse sicherheitskritischer Systeme und Anwendungen. In *DFN-Forum Kommunikationstechnologien*, pages 87–96, 2013.

[GS11] Mario Golling and B Stelte. Requirements for a Future EWS-Cyber Defence in the Internet of the Future. In *Cyber Conflict (ICCC), 2011 3rd International Conference on*, pages 1–16. IEEE, 2011.

[KGH11] Michael Kretzschmar, Mario Golling, and Sebastian Hanigk. Security Management Areas in the Inter-Cloud. In *Cloud Computing (CLOUD), 2011 IEEE International Conference on*, pages 762–763. IEEE, 2011.

[KGSR] Robert Koch, Mario Golling, Lars Stiemert, and Gabi Dreo Rodosek. Using Geolocation for the Strategic Pre-Incident Preparation of an IT Forensics Analysis. *IEEE Systems Journal*. to appear.

[KSG12] R. Koch, B. Stelte, and M. Golling. Attack Trends in Present Computer Networks. In *Cyber Conflict (CYCON), 2012 4th International Conference on*, pages 1–12. IEEE, 2012.

[MCC+04] RL Morgan, Scott Cantor, Steven Carmody, Walter Hoehn, and Ken Klingenstein. Federated Security: The Shibboleth Approach. *Educause Quarterly*, 27(4):12–17, 2004.

[NJM+12] Shirin Nilizadeh, Sonia Jahid, Prateek Mittal, Nikita Borisov, and Apu Kapadia. Cachet: A Decentralized Architecture for Privacy Preserving Social Networking with Caching. In *Proceedings of the 8th international conference on Emerging networking experiments and technologies*, pages 337–348. ACM, 2012.

[PJN10] Emmanuel S Pilli, Ramesh C Joshi, and Rajdeep Niyogi. Network Forensic Frameworks: Survey and Research Challenges. *Digital Investigation*, 7(1):14–27, 2010.

[Ren04] Wei Ren. On A Reference Model of Distributed Cooperative Network, Forensics System. In *iiWAS*, 2004.

[Sys11] Cisco Systems. *Cisco Security Appliance System Log Messages Guide*, 2011.

[TD05] Yongping Tang and Thomas E Daniels. A Simple Framework for Distributed Forensics. In *Distributed Computing Systems Workshops, 2005. 25th IEEE International Conference on*, pages 163–169. IEEE, 2005.