Eine Sicherheitsarchitektur für offene, lokale Netze. Realisierung durch Open Source Software?

Prof. Dr. Josef von Helden, Prof. Dr. Stefan Wohlfeil

Fachbereich Informatik
Fachhochschule Hannover
Ricklinger Stadtweg 120
30459 Hannover
[josef.vonhelden|stefan.wohlfeil]@inform.fh-hannover.de

Abstract: Durch die bedarfsgetriebene Öffnung lokaler Netze für Fremdgeräte nehmen Sicherheitsrisiken erheblich zu. Im vorliegenden Dokument wird eine Grundstruktur für Sicherheitsarchitekturen in offenen, lokalen Netzen vorgestellt, die durch die Kombination mehrerer Sicherheitskomponenten einen signifikanten Sicherheitsgewinn verspricht. Die TNC-Architektur der Trusted Computing Group basiert auf diesem Ansatz. Die Autoren zeigen auf, wie Open Source Software hierbei eingesetzt werden kann.

1 Einleitung

Sicherheitsmaßnahmen in Netzen konzentrieren sich heute zu einem großen Teil auf den Übergang zwischen lokalen, vertrauenswürdigen und fremden, nicht-vertrauenswürdigen Netzen. Dort reduzieren meist Firewall-Techniken die Risiken auf ein akzeptables Maß. Das lokale Netz hält man für vertrauenswürdig und verzichtet auf weitere Schutzmaßnahmen. Deshalb kann jedes Endgerät im Netz i.d.R. alle absichtlich oder unkontrolliert angebotenen Dienste erreichen. Häufig will man auch Fremdgeräte - d.h. Endgeräte, die nicht unter der administrativen Kontrolle des Netzbetreibers stehen und die daher nicht vertrauenswürdig sind - im lokalen Netz zulassen. Beispiele hierfür sind notebooks oder PDAs von Studierenden in Hochschulen oder externe Berater mit eigenem notebook in Unternehmen sowie auch privat genutzte Endgeräte von MitarbeiterInnen.

Da die Vertrauenswürdigkeit der Endgeräte also nicht mehr vorausgesetzt werden kann, muss das Netz entsprechende Schutzvorkehrungen treffen. Ein Ansatz hierzu ist, die Vertrauenswürdigkeit der Endgeräte detaillierter einzuschätzen und entsprechend abgestufte Maßnahmen bei ihrer Integration zu treffen.

2 Sicherheitsarchitekturen für offene, lokale Netze

Es gibt bereits heute eine Vielzahl von Sicherheitsstandards, -maßnahmen, -technologien und -produkte, die nicht nur zwischen Netzen sondern auch innerhalb lokaler Netze Bedeutung haben. Hierzu gehören Standards wie IEEE 802.1x, Filtertechniken und

ACLs (Access Control Lists) in Routern, Intrusion Detection und Prevention Systeme auf Servern, Sensoren und in aktiven Netzkomponenten sowie Virenschutzmaßnahmen.

Ein signifikanter Sicherheitsgewinn ließe sich durch ein strukturiertes Zusammenspiel solcher Sicherheitskomponenten im Sinne einer Sicherheitsarchitektur erzielen. Kennt man alle im Netz angekoppelten Endgeräte, inkl. ihrer Eigenschaften , dann können sie kontrolliert zugelassen oder abgelehnt werden. In Abschnitt 2.1 werden Kernfunktionen einer Sicherheitsarchitektur für offene, lokale Netze skizziert. Im darauf folgenden Abschnitt 2.2 wird auf die TNC-Architektur der Trusted Computing Group [TN05] eingegangen, die wegen ihrer Offenheit mehr Möglichkeiten für die Entwicklung und den Einsatz von Open Source Software bietet als vergleichbare, proprietäre Ansätze wie „Network Admission Control“ (NAC) von Cisco[1].

2.1 Kernfunktionen

Abbildung 1 zeigt die Kernfunktionen einer Sicherheitsarchitektur. Dabei sind für jeden Schritt obligatorische (MUSS) und optionale (KANN) Anforderungen an das Netz - d.h. im Wesentlichen an die Software in aktiven Netzkomponenten und Servern - aufgeführt.

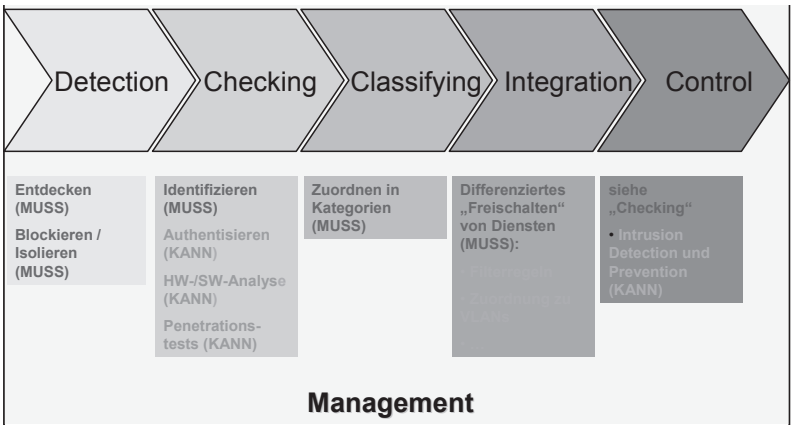

Abbildung 1: Kernfunktionen der Sicherheitsarchitektur

2.2 TCG Trusted Network Connect

Die Trusted Network Connect (TNC) Arbeitsgruppe der Trusted Computing Group (TCG) hat im Mai 2005 eine offene Architektur für die Authentisierung und Integritätsprüfung von Endgeräten in Netzen vorgeschlagen und spezifiziert [TN05].

Abbildung 2 zeigt den Ablauf der Überprüfung eines Endgerätes, hier eines Laptops, bei seiner Ankopplung an ein TNC-Netz. Der Laptop (Access Requestor, AR) wird über

[1] Siehe auch http://www.cisco.com/en/US/netsol/ns466/networking_solutions_package.html

einen Policy Enforcement Point (PEP), typischerweise einen Switch, an das Netz angeschlossen. Ein Policy Decision Point (PDP) entscheidet, was der Laptop im Netz nun darf und was nicht.

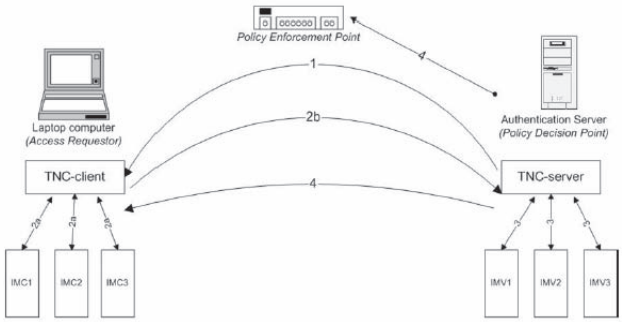

Abbildung 2: Ankopplung eines Endgerätes mit Überprüfung in TNC [TN05c]

Der Ablauf der Überprüfung besteht aus folgenden Schritten:

1. Der TNC-server fordert den TNC-client auf, seine Identitäts- und Integritätsinformationen bereit zu stellen.

2a. Der TNC-client sammelt diese Informationen von seinen IMCs (Integrity Measurement Collector).

2b. Der TNC-client schickt diese Informationen über den PEP an den TNC-server.

3. Der TNC-server überprüft mit Hilfe seiner IMVs (Integrity Measurement Verifier) die empfangenen Informationen.

4. Der TNC-server trifft - auf der Basis der jeweiligen Sicherheitsrichtlinien - eine Entscheidung über Zulassung, Ablehnung oder teilweiser Integration des ARs und teilt diese Entscheidung dem PEP und dem AR mit.

Bei diesem Architekturansatz ist sicherlich zu hinterfragen, welche Formen der Kontrolle eines Endgerätes durch ein Netz technisch machbar sowie gegenüber einem Benutzer, der für Datenschutz sensibilisiert ist, vertretbar sind[2]. Die TCG verweist zu diesem Thema auf die Möglichkeit der Integration seiner vertrauenswürdigen Hardware-Komponente „Trusted Platform Module (TPM)“ [TC04].

3 Einsatz von Open Source Software

3.1 Ausgangssituation

Derzeit ist Open Source Software auf dem Markt, die für manche Security-Aufgaben bestens geeignet ist. Ein wesentlicher Sicherheitsgewinn ist zu erwarten, wenn die unter-

[2] Hierzu findet auch im Zusammenhang mit dem Thema „Trusted Computing“ eine intensive Debatte statt [An03].

schiedlichen Software-Komponenten und -Technologien im Sinne einer strukturierten Sicherheitsarchitektur zu einem kooperierenden Software-System zusammen wachsen. Da die Sicherheitsarchitektur TNC durch ihre offen gelegten Spezifikationen die grundsätzliche Voraussetzung sowohl für die Entwicklung von Open Source Software als auch für Interoperabilität bietet, wird hierauf im folgenden Abschnitt genauer eingegangen.

3.2 Open Source-Entwicklung im Rahmen von TNC

Abbildung 3 zeigt die TNC-Architektur mit ihren Schnittstellen.

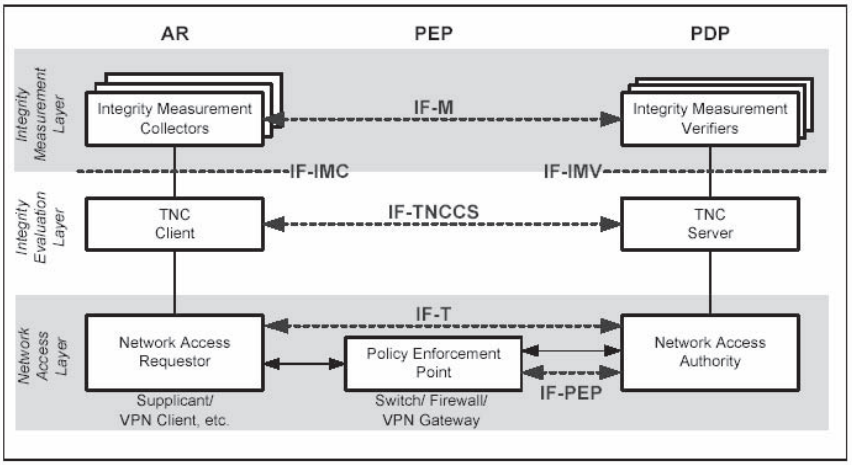

Abbildung 3: Die TNC-Architektur [TN05]

Die beiden, bereits spezifizierten Schnittstellen IF-IMC [TN05a] und IF-IMV [TN05b] bilden die Grundlage für eine interoperable Kommunikation zwischen TNC-Clients und herstellerspezifischen IMCs auf der AR-Seite und zwischen TNC-Servern und herstellerspezifischen IMVs auf der PDP-Seite.

Gute Möglichkeiten für Open Source Implementierungen werden vor allem in folgenden Bereichen gesehen:

- Entwicklung von IMCs und IMVs für Open Source Werkzeuge, die Auskunft über den Sicherheitsstatus und die Integrität von Endgeräten geben, z.B. Virenscanner wie *ClamWin*, Integritätsschutz-Software wie *tripwire* oder Rootkit Detection Tools wie *RootkitRevealer*.
- Integration von Open Source Software, die im Netz ohne kooperierende Software auf dem Endgerät läuft. Hierzu zählen beispielsweise Schwachstellen-Scanner wie *nessus*, Intrusion Detection Werkzeuge wie *snort* und Netzwerkmonitore wie *ethereal*.
- Open Source-Implementierung von TNC-Clients, -Servern und ggf. weiterer TNC-Komponenten, z.B. auf Linux-Basis.

4 Fazit

Mit einer wohl strukturierten Sicherheitsarchitektur kann durch die Kooperation unterschiedlicher sicherheitsrelevanter Komponenten die Sicherheit in offenen, lokalen Netzen signifikant erhöht werden. Die TNC-Architektur der TCG erscheint aufgrund ihrer offenen Spezifikationen und dem Ziel der Interoperabilität als gute Ausgangsbasis für die Integration und die Entwicklung von Open Source Software. Herausforderungen und Fragestellungen für die Zukunft werden vor allem sein:

- In wie weit können Open Source Werkzeuge als zuverlässige und stabile Bestandteile in eine solche Sicherheitsarchitektur integriert werden?
- Wie findet man die richtige Balance zwischen Sicherheitsansprüchen und dem Schutz der Privatsphäre?
- Kann man auf einfache Weise geeignete Sicherheitsrichtlinien definieren, die durch die Sicherheitsarchitektur möglichst automatisch durchgesetzt werden?

5 Quellenverzeichnis

[An03] Ross Anderson; `Trusted Computing' Frequently Asked Questions - TC / TCG / LaGrande / NGSCB / Longhorn / Palladium / TCPA; Version 1.1 (August 2003); http://www.cl.cam.ac.uk/~rja14/tcpa-faq.html

[TC04] TCG Specification, Architecture Overview, Specification, Revision 1.2, 28 April 2004 https://www.trustedcomputinggroup.org/downloads/TCG_1_0_Architecture_Overview.pdf

[TN05] TCG Trusted Network Connect, TNC Architecture for Interoperability Specification Version 1.0, Revision 4, 3 May 2005, https://www.trustedcomputinggroup.org/downloads/specifications/TNC_Architecture_v1_0_r4.pdf

[TN05a] TCG Trusted Network Connect, TNC IF-IMC, Specification Version 1.0, Revision 3, 3 May 2005, https://www.trustedcomputinggroup.org/downloads/specifications/TNC_IFIMC_v1_0_r3.pdf

[TN05b] TCG Trusted Network Connect, TNC IF-IMV, Specification Version 1.0, Revision 3, 3 May 2005 https://www.trustedcomputinggroup.org/downloads/specifications/TNC_IFIMV_v1_0_r3.pdf

[TN05c] Trusted Network Connect, Open Standards for Integrity-based Network Access Control, 2005 https://www.trustedcomputinggroup.org/downloads/whitepapers/Open_Standards_for_IntegrityBased_AccessControl.pdf