Autokonfiguration für Kommunikationssicherheit

Lars Völker, Christoph Werle, Marcel Noe, Benjamin Behringer
Institut für Telematik, Universität Karlsruhe (TH), Germany

Steckbrief

Kategorie:	Software aus einem Forschungsprojekt an einer Hochschule
Funktion:	Automatischer Kommunikationsschutz
Höhepunkte:	• Automatisch erhöhte Sicherheit • Hohe Nutzerfreundlichkeit • Trade-Off: Sicherheit vs. Kosten • Strategie für zukünftige Protokolle
Plattform:	Vollständiges System: • GNU Linux 2.6 (derzeit) Nur HTTP-Browser-Add-on: • Mozilla Firefox, plattformunabhängig
Verfügbarkeit:	Die Konzepte, Ergebnisse und Quelltext werden frei verfügbar sein.

I. Einleitung

Heutige Sicherheitsprotokolle für die Nutzung im Internet überfordern durchschnittliche Anwender oft. Eine Hürde bei der Verwendung von Sicherheitsprotokollen besteht schon in der Auswahl und Konfiguration eines Protokolles. Wird schließlich ein Sicherheitsprotokoll verwendet, so werden Benutzer auch zur Laufzeit oft mit für sie unverständlichen Fragen, z.B. zur Überprüfung eines Zertifikates, und Fehlermeldungen, belästigt [1].

Statt den Nutzer zu stark zu fordern, sollten die Protokolle vielmehr im Hintergrund für den Nutzer arbeiten. Daher ist statt der nutzerbasierten Wahl zwischen mehreren Protokollen und deren Konfiguration eine qualifizierte Auswahl des Protokolls und Konfiguration deutlich vorteilhafter.

Die hier vorgestellte Software *Autokonfiguration für Kommunikationssicherheit (AKKS)* hat genau dies zum Ziel: Die Auswahl und Konfiguration von Sicherheitsprotokollen für den zu Nutzer übernehmen und ihn dadurch zu entlasten.

Hierbei gilt als Randbedingung, dass dieser Schutz bereits im heutigen Internet funktionieren soll, also keine Änderungen an anderen Systemen im Internet notwendig sind. Nur so kann erreicht werden, dass in sehr kurzer Zeit eine wesentliche Verbesserung für den Nutzer erreicht wird.

Mit Hilfe dieser Strategie kann zugleich auch erreicht werden, dass die Nutzung von zukünftigen Sicherheitsprotokollen vereinfacht wird, da diese durch AKKS automatisch gewählt und genutzt werden können. Bislang mussten Anwendungen neue Sicherheitsprotokolle selbst auswählen, was zur Notwendigkeit von Updates der Anwendungen führte.

Im Folgenden wird der AKKS-Demonstrator sehr kurz beschrieben, sowie zwei interessante Aspekte des Demonstrators hervorgehoben.

II. Demonstrator

Der hier vorgestellte AKKS-Demonstrator hat zum Ziel die Auswahl und Konfiguration von Sicherheitsprotokollen für den Nutzer übernehmen und ihn dadurch zu entlasten. Eine schematische Darstellung des Demonstrators erfolgt in Abbildung 1.

Die zentrale Komponente von AKKS ist der *Sicherheitsmanager*, welcher neue Kommunikationsassoziationen auf Schutzbedarf und Schützbarkeit untersucht. Hierauf basierend kann er dann ein Schutzverfahren, in der Regel ein Sicherheitsprotokoll, auswählen und konfigurieren. Die Entscheidungsfindung des Sicherheitsmanager wird detailliert in Abschnitt II-A beschrieben.

Der Zugriff auf die Sicherheitsprotokolle wird durch *Adapter* erreicht. Diese Adapter beantworten die Anfrage des Sicherheitsmanagers nach Schützbarkeit einer gegebenen Kommunikationsassoziation. Hierbei muss der Adapter basierend auf der Beschreibung der Kommunikationsassoziation feststellen, ob diese durch das von ihm repräsentierte Protokoll geschützt werden kann. Dabei können von einem Adapter auch mehrere Alternativen zur Verfügung gestellt werden, welche z. B. durch die Existenz verschiedener Cipher Suites eines Sicherheitsprotokolls gegeben sind. Diese werden dem Sicherheitsmanager von den Adaptern als unterschiedliche Optionen angeboten. Wichtig hierbei ist, dass sich diese Optionen mit Hinblick auf verschiedenen Eigenschaften, wie beispielsweise Sicherheit, QoS und Energiebedarf, unterscheiden können. In Abschnitt II-B werden Adapter und Möglichkeiten für ein ausgewähltes Protokoll, das Hypertext Transfer Protokoll (HTTP), beschrieben.

Im Folgenden gehen wir auf einige bemerkenswerte Teile des Demonstrators ein.

A. Entscheidungsfindung

Die im Sicherheitsmanager implementierte Entscheidungsfindung [2] basiert auf der Multi-Attribute Utility Theory (MAUT). Sie ermöglicht es den optimalen Trade-Off zwischen Wirkung und Nebenwirkungen eines Sicherheitsprotokolls zu wählen. Die *Wirkungen* eines Sicherheitsprotokolls umfassen die spezifischen Sicherheitsauswirkungen auf die Kommunikation. Hierzu gehören Authentizitätsschutz, Geheimhaltung und vieles mehr. Die *Nebenwirkungen* der Sicherheitsprotokolle umfassen andere, weniger positive Auswirkungen, wie z. B. erhöhte Latenz, verringerte mögliche Bandbreite und erhöhten Stromverbrauch.

Unsere Implementierung der Entscheidungsfindung zeichnet sich dadurch aus, dass durch mehrere Optionen Einfluß auf den Prozeß der Entscheidungsfindung genommen werden kann.

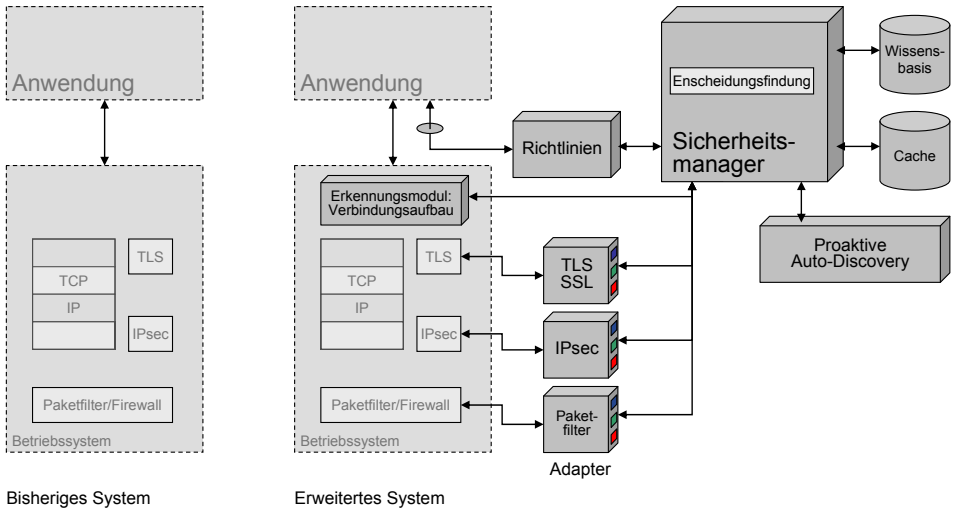

Abbildung 1. Der Vergleich eines bisherigen System zu einem System mit AKKS

B. Schutz von HTTP

Versucht man, automatisch beliebige Kommunikationsassoziationen zu schützen, stößt man bei einigen Protokollen auf Schwierigkeiten oder interessante Verbesserungsmöglichkeiten. Gerade das Hypertext Transport Protokoll (HTTP) ist ein solcher Fall. Das geschützte Pendant zu HTTP ist Hypertext Transport Protocol Secure (HTTPS). Nun könnte man annehmen, dass durch die Existenz von HTTPS der automatische Schutz von HTTP relative einfach wäre. Dies ist allerdings nicht der Fall, da sich der Inhalt einer Webseite, abhängig davon, ob sie über HTTP oder HTTPS abgerufen wird, unterscheiden kann. Zwei Besonderheiten im AKKS-Demonstrator gehen auf HTTP-Eigenarten ein:

1. Die *Gleichheitsüberprüfung der über HTTP und HTTPS übertragenen Inhalte* ist eine direkte Konsequenz der Besonderheiten von HTTP. Durch ein spezialisiertes Vergleichsverfahren kann auch bei dynamisch erzeugten Inhalten die Gleichheit festgestellt werden.

2. Mit der *Integration in den Browser* kann ein Teil von AKKS bereits sehr leicht genutzt werden. Dazu wird dem Browser mittels Add-On ein vereinfachtes Vergleichsverfahren und eine vereinfachte Entscheidungsfindung bereitgestellt. Dies erlaubt einen gewissen Teilerfolg ohne die vollständige Installation von AKKS. Weitere Verbesserungen werden durch Kopplung an das AKKS-System erreicht. Durch diese Integration von AKKS und Browser können dem Nutzer zudem Informationen über die Sicherheit seiner derzeitigen Verbindung gegeben werden.

III. Zusammenfassung und Ausblick

Der hier kurz vorgestellte Demonstrator ermöglicht die automatische Wahl von Sicherheitsprotokollen. Dies führt zu erhöhter Sicherheit und verbesserter Nutzerfreundlichkeit. Zusätzlich wird die Integration neuer Sicherheitsprotokolle vereinfacht, da deren die Wahl von Anwendungen unabhängig wird.

Die weitere Evolution des Demonstrators wird die Integration weiterer Sicherheitsprotokolle umfassen, sowie Optimierungen, schnellere und genauere Entscheidungen treffen zu können.

Der Demonstrator erreicht bereits für das heutige Internet deutliche Verbesserungen, ohne dabei zu verlangen, dass Systeme im Internet geändert werden. Hierdurch wird die Sicherheit des Nutzers alleinig durch die Modifikation des eigenen Systems erreicht. Es bleibt dennoch notwendig festzustellen, ob diese Bedingungen im zukünftigen Internet gelten oder ob bessere Alternativen existieren könnten. Als Ausblick verweisen wir daher auf unsere Architektur für das zukünftige Internet [3], welche die Konzepte von AKKS auf ein Future Internet mit Clean-Slate-Ansatz anwendet. Hierbei wird dann nicht nur die automatische Wahl von Sicherheitsprotokollen möglich sein, sondern auch die Wahl zwischen verschiedensten Protokollen basierend auf Anforderungen und Eigenschaften, wie Sicherheit, QoS und Energieverbrauch.

Literatur

[1] A. Whitten and J. Tygar, “Why Johnny can’t encrypt: A usability evaluation of PGP 5.0”, *8th USENIX Security Symposium*, 1999.

[2] L. Völker, C. Werle, and M. Zitterbart, “Decision Process for Automated Selection of Security Protocols”, in *Proceedings of the 33nd IEEE Conference on Local Computer Networks (LCN 2008)*. Montreal, QB, Canada: IEEE, Oct. 2008, pp. 223–229.

[3] L. Völker, D. Martin, I. El Khayat, C. Werle, and M. Zitterbart, “An Architecture for Concurrent Future Networks”, in *2nd GI/ITG KuVS Workshop on The Future Internet*. Karlsruhe, Germany: GI/ITG Kommunikation und Verteilte Systeme, Nov. 2008.

[4] L. Völker, M. Schöller, and M. Zitterbart, “Introducing QoS mechanisms into the IPsec packet processing”, in *Proceedings of the 32nd IEEE Conference on Local Computer Networks (LCN 2007)*. Dublin, Ireland: IEEE, Oct. 2007, pp. 360–367.

[5] L. Völker and M. Schöller, “SecureTLS: Preventing DoS Attacks with Lower Layer Authentication”, in *Kommunikation in Verteilten Systemen (KiVS) 2007*, T. Braun, G. Carle, and B. Stiller, Eds. Bern, Switzerland: Springer, Feb. 2007, pp. 235–248.