Eine Software-Architektur zur Konstruktion flexibler IPsec-Infrastrukturen

Michael Rossberg[1], Wolfgang Steudel[1], Günter Schäfer[1] und Kai Martius[2]

Kurzfassung:
VPN (virtuelle private Netzwerke) auf Basis von IPsec bieten einen weitreichenden Schutz gegen Angriffe auf die Vertraulichkeit und Integrität übertragener Daten. Durch ihre komplexe und oft statische Konfiguration wird allerdings die Flexibilität in Bezug auf Integration neuer IPsec-Gateways oder die Anpassung von Routen im VPN stark eingeschränkt.
Ausgehend von einem dezentralen Ansatz zu IPsec-Autokonfiguration [RSS09] wird in diesem Beitrag eine Software-Architektur vorgestellt, die es zum einen erlaubt, in einem Simulator den Konfigurationsmechanismus sowie die entstehenden VPN-Topologien in Bezug auf Robustheit sowie weitere Leistungsmerkmale zu bewerten, und die zum anderen auch eine Implementierung des Ansatzes auf Basis von Linux und strongSWAN bereitstellt.

Stichworte: Sichere Kommunikationsprotokolle, Mobile Plattformen und Schnittstellen, Infrastruktur-Sicherheit, Betriebssysteme, Entwicklungskonzepte

1. Einleitung und Motivation

Virtuelle private Netzwerke (VPN) ermöglichen eine transparente und zugleich sichere Kommunikation zwischen geographisch verteilten Partnern über nicht-vertrauenswürdige Netze. Allerdings erfolgt die Konfiguration der dafür häufig eingesetzten IPsec-Infrastrukturen in der Regel manuell. Das bedeutet es werden selbst bei großen Netzwerken zwischen den beteiligten IPsec-Gateways paarweise Sicherheitsbeziehungen eingerichtet. Durch den enormen Aufwand, welcher mit der Anzahl der IPsec-Gateways quadratisch wächst, ist dieses Verfahren kostenintensiv und potenziell fehleranfällig. Ein weiteres Defizit ist die fehlende Flexibilität, da Veränderungen in der VPN-Topologie immer einen manuellen Eingriff erfordern.

Ansätze zur automatischen Rekonfiguration von IPsec-Infrastrukturen, wie Ciscos Tunnel Endpoint Discovery (TED) [Cis05], Group Encrypted Transport VPN (GET) [Bah08], und das Policy Framework for IP Security der IETF [SS99], konstruieren die VPN-Topologie auf geschickte Weise, sodass die Security Policy Database (SPD) in den einzelnen IPsec-Gateways relativ statisch konfiguriert werden kann. Leider haben diese Verfahren einige system-immanente Nachteile. Sie können beispielsweise nicht die Routing-Funktionalität eines manuell konfigurierten IPsec-VPN nachbilden, sodass der Einsatz in Szenarien mit geschachtelten, sicheren Netzwerken oder bei der Verwendung von privaten IP-Adressen in den sicheren Netzwerken nicht möglich ist.

Aus Sicht der Sicherheit ist aber gerade die Identität eines IPsec-Gateways und dessen „Autorität“ (für welche angeschlossenen Netze er zuständig ist) relevant, nicht jedoch dessen Erreichbarkeit in einem Transportnetz (Tunnelendpunkt). Daher ist es wün-

1 Technische Universität Ilmenau, Fachgebiet Telematik/Rechnernetze, Ilmenau
2 secunet Security Networks AG, Dresden

schenswert, die Erreichbarkeit des Gateways von seiner Sicherheitskonfiguration zu trennen, indem beispielsweise private IP-Adressbereiche für sein inneres Teilnetz verwendet werden. Diese Eigenschaft kann mit den erwähnten Ansätzen somit nicht erreicht werden.

Daher wird in diesem Artikel SOLID (Secure OverLay for IPsec Discovery) vorgestellt, dessen Softwarearchitektur eine dynamische Anpassung der SPD je nach Netzwerkgegebenheiten vornehmen kann. Durch die gewonnene Flexibilität kann ein Hintergrundprogramm in den IPsec-Gateways eine agile Topologiekontrolle durchführen und das VPN so dynamisch an die Gegebenheiten im Netzwerk anpassen.

Im Folgenden werden zunächst Anforderungen an eine Rahmenarchitektur zur flexiblen Konfiguration von IPsec VPN gegeben, bevor die zugrunde gelegte Netzwerkstruktur und die SOLID-Komponenten in Abschnitt 3 und 4 erläutert werden. Anschließend erfolgen eine Bewertung des Ansatzes (Abschnitt 5) und eine Zusammenfassung (Abschnitt 6). Der Artikel schließt mit einem Überblick über geplante Entwicklungen in Abschnitt 7.

2. Anforderungen an die Software-Architektur

Der dynamische Aufbau von IPsec-VPN Strukturen erfordert eine Erweiterung herkömmlicher Schlüssel-Management-Verfahren um ein System mit folgenden Eigenschaften:

Flexibilität: Die Softwarearchitektur muss Veränderungen im VPN und Transportnetzwerk dynamisch erfassen und gegebenenfalls auf Änderungen reagieren und so beispielsweise Pfade durch das VPN bei Bedarf umleiten können. So kann den Auswirkungen von Denial-of-Service-Angriffen begegnet werden.

Reaktives Verhalten: In sehr dynamischen VPN ist es nicht mehr möglich Sicherheitsbeziehungen zwischen allen IPsec-Gateways aufzubauen, da in diesem Fall die Veränderung der Netzwerkkonnektiviät eines Gateways eine Aktualisierung aller anderen bewirken würde. Daher sollte es die Möglichkeit geben, IPsec-Beziehungen nur bei Bedarf aufzubauen.

Sicherheit: Durch die gewonnene Flexibilität dürfen die durch IPsec gegebenen Sicherheitseigenschaften, wie die Vertraulichkeit und die Integrität der übertragenen Daten, nicht verletzt oder reduziert werden.

Transparenter Betrieb: Da eine Modifikation der Systeme innerhalb der sicheren Netzwerke nicht zu vertreten ist, muss die Softwarearchitektur Rekonfigurationsvorgänge des VPN vor diesen Endsystemen verbergen.

Integrierbarkeit: Existierende Softwarekomponenten, wie die IPsec-Implementierung im Betriebssystemkern, sind nach Möglichkeit nicht zu verändern, da sonst eine Pflege der Software sehr aufwendig werden kann. Insbesondere wird so eine erneute Zertifizierung der sicherheitskritischen Komponenten vermieden.

Simulierbarkeit: Der Test eines dynamischen Konfigurationssystems in großen VPN-Szenarien ist extrem aufwendig, daher sollen wesentliche Teile der Software in Simulationen zu evaluieren sein und so bereits vor jedem größeren Roll-out mögliche Komplikationen entdeckt werden können.

Die hier vorgestellten Anforderungen sind größtenteils aus der Analyse praktischer Einsatzumgebungen beziehungsweise neuer Kundenanforderungen zusammengeführt worden. Die Erfahrungen aus der Installation und dem Betrieb einiger tausend VPN-Komponenten zeigen, dass in Netzen mit privaten IP-Adressen, der Forderung nach Bildung unterschiedlicher Sicherheitsdomänen sowie generell komplexe Netzstrukturen IPsec-VPNs aktuell einen hohen Managementaufwand verursachen. Viele Kunden scheuen noch die zusätzliche Komplexität durch das VPN-Overlay-Netz und orientieren teilweise auf andere Sicherheitsmaßnahmen (Layer-2-Verschlüsselung), die nicht die Funktionalität und Ende-zu-Ende-Sicherheit einer IPsec-VPN-Lösung bieten. Neue Anforderungen aus dem militärischen Umfeld für den Einsatz von IPsec im taktisch-mobilen Bereich („Network Enabled Capabilities“) zeigen weiteren Bedarf an einer einfach und sicher zu verwaltenden Lösung.

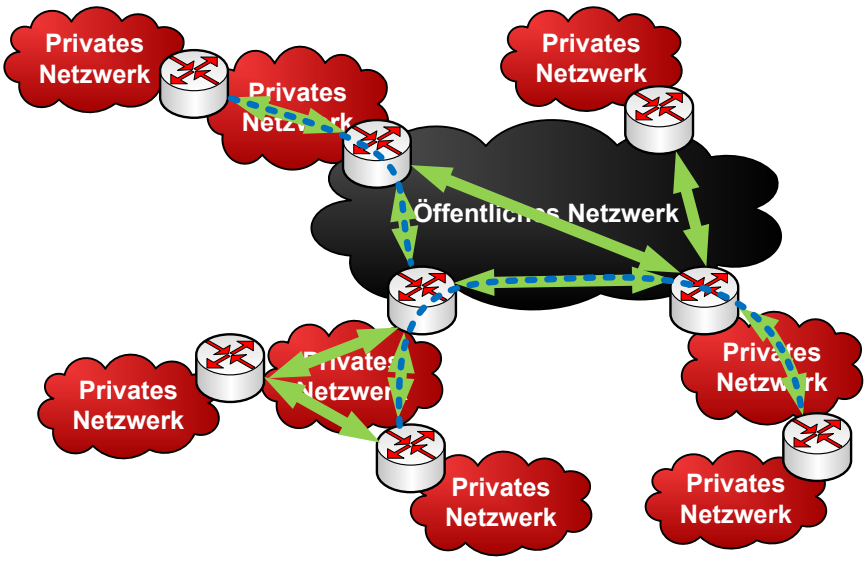

Abbildung 1 – Overlay-Schichten in SOLID

3. Netzwerk-Architektur

VPN-Topologien können teilweise sehr komplex werden. Abbildung 1 zeigt ein Beispielnetzwerk, bei dem verschiedene private Netzwerke über ein öffentliches Netzwerk miteinander kommunizieren. Im Gegensatz zu vielen manuell konfigurierten VPN, bei denen IPsec-Tunnelverbindungen geschachtelt werden, um eine Routing-Funktionalität zu erreichen, werden von SOLID zunächst nur zwischen direkt benachbarten IPsec-Gateways Beziehungen aufgebaut. Sollen Gateways kommunizieren, die über keine direkte Verbindung verfügen, so wird von SOLID genau ein geschachtelter Tunnel aufgebaut, welcher der Ende-zu-Ende-Sicherung dient. In den Zwischensyste-

men erfolgt eine Weiterleitung an Hand von virtuellen Pfaden, die ebenfalls automatisch von SOLID eingerichtet werden.

Die virtuellen Pfade entlang der SOLID-Gateways erlauben lokale Reparaturen, um schnell auf Veränderungen im Transportnetz reagieren zu können. Gleichzeitig ist durch die Beschränkung auf eine einfache Verschachtelung eine konstante maximale Paketgröße („MTU“) gewährleistet, sodass eine aufwendige Fragmentierung in Zwischensystemen verhindert werden kann.

In herkömmlichen IPsec-VPN werden Beziehungen im Tunnelmodus verwendet, um Pakete von Client-Geräten weiterzuleiten. Im Gegensatz dazu wird in SOLID ein Verfahren verwendet, das IIPtran [TE04] ähnelt. Das Verfahren, wie in **Abbildung 2** dargestellt, sieht zwischen benachbarten Gateways IPsec-Beziehungen im Bound-End-to-End-Tunnel (BEET) Modus [NM08] vor. Dieser verhält sich prinzipiell wie eine reguläre IPsec-Transport-Beziehung, jedoch wird bei der IKE-Aushandlung zusätzlich ein privates IP-Adresspaar ermittelt, welches der semantischen Identifikation des jeweils anderen Gateways dient. Durch die strikte Trennung von Identität und Erreichbarkeit wird eine Reihe von Problemen bisheriger IPsec-Ansätze gelöst. So vereinfacht sich das Zusammenspiel mit nicht-vertrauenswürdigen Routern, die Network Address Translation (NAT) verwenden, und der Wechsel der externen IP-Adressen von Sicherheits-Gateways führt nicht zwangsläufig zu einem Bruch der Sicherheitsbeziehung.

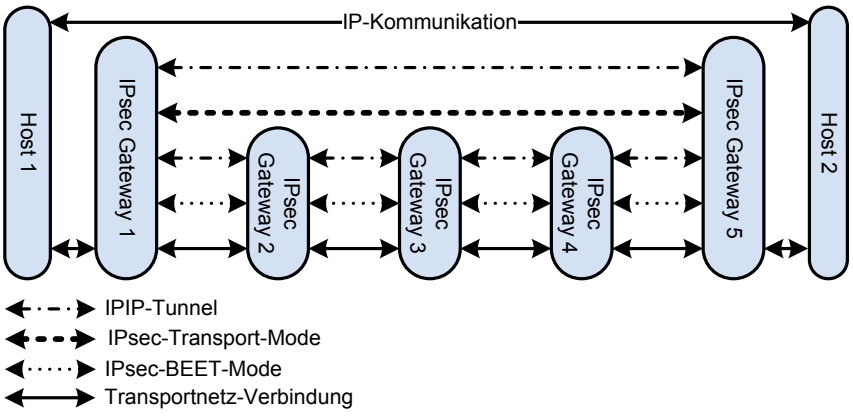

Abbildung 2 – Overlay-Schichten in SOLID

Um Pakete zwischen benachbarten Gateways weiterleiten zu können, werden von SOLID IPIP-Tunnel eingesetzt, sodass die übertragenen Pakete nach der IPsec-BEET-Verarbeitung die gleiche Struktur wie Pakete haben, die durch den IPsec-Tunnel-Modus geschützt werden. Neben einem geringeren Protokoll-Overhead, der durch das Weglassen der IPIP-Header bei der direkten Kommunikation zwischen zwei Gateways entsteht, wird so eine strikte Trennung zwischen Routing und IPsec erreicht. Daraus resultiert eine größere Flexibilität, da beispielsweise weiterzuleitende Pakete im Fehlerfall einfach über eine andere IPsec-Beziehung versandt werden können. Zu beachten ist hierbei allerdings, dass die Bindung eines IP-Adressbereiches an eine IPsec-Beziehung zunächst nicht gegeben ist. Das bedeutet, jedes Gateway kann prinzipiell

beliebige innere IP-Adressen verwenden. Dynamische Firewall-Regeln in den Sicherheits-Gateways, die beim Anlegen der IPIP-Tunnel eingerichtet werden, verhindern dies, so dass die gleichen Sicherheitsziele wie beim IPsec-Tunnelmodus erreicht werden können.

Für Verbindungen zwischen IPsec-Gateways, die über keine direkte Transportnetzverbindung zueinander verfügen, verwendet SOLID eine weitere, geschachtelte IPsec-Transport-Beziehung sowie dazu gehörigen IPIP-Tunnel, um die Ende-zu-Ende-Sicherheit zu gewährleisten.

4. Die Software-Architektur von SOLID

Die SOLID-Architektur (siehe **Abbildung 3**) implementiert innerhalb eines Standard-Linux-Systems dynamisch eine VPN-Topologiekontrolle und Routing. Linux wurde unter anderem deshalb gewählt, weil es sich durch seine Quelloffenheit und umfangreiche Netzwerk- und IPsec-Funktionalität auszeichnet. Eine weitere Hauptkomponente, mit der SOLID interagiert, ist der Charon-Dienst, welcher im Rahmen des strongSWAN-Projektes entwickelt wird und das IKEv2-Schlüsselaustauschprotokoll für IPsec [Kau05] implementiert.

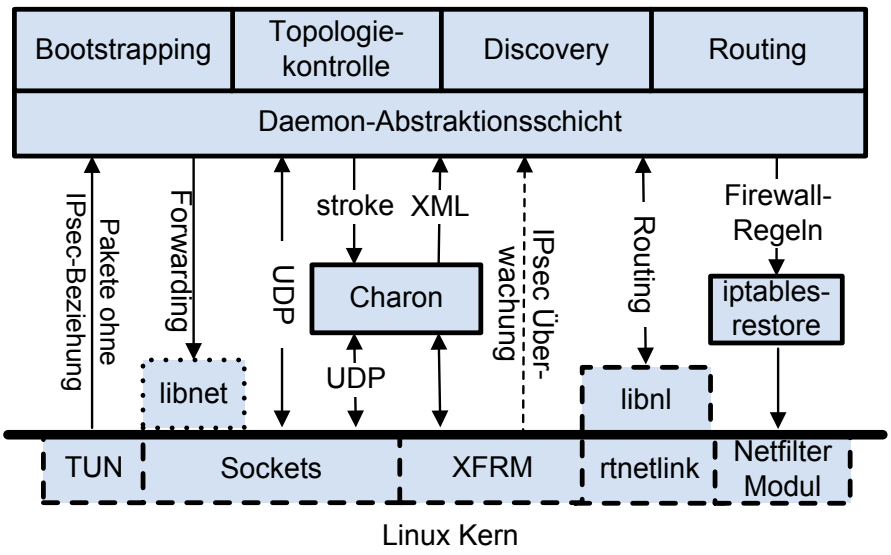

Abbildung 3 – SOLID Linux-Integration

Der SOLID-Daemon nutzt diese Umgebung, um ein dynamisches VPN-Overlay vollständig autonom aufzubauen. Dazu initiiert die SOLID-Topologiekontrolle nach dem Start zunächst einen Bootstrapping-Prozess, um weitere aktive SOLID-Gateways zu finden. Je nach Netzwerkumgebung kommen verschiedene Mechanismen zum Einsatz. Beispielsweise werden in Local Area Networks Broadcast-Pakete verwendet, während in Weitverkehrsnetzen DNS-Anfragen verwendet werden.

Anschließend kann SOLID vollautomatisch eine Sicherheitsassoziation zu den gefundenen Gateways erzeugen. Über das proprietäre stroke-Interface wird im Charon-Daemon ein Security-Policy-Database-Eintrag generiert und ein IKEv2-Austausch veranlasst. Bei erfolgreichem Aufbau der IPsec-Beziehung werden von Charon IPsec-

Policies und SAs über die XFRM-Schnittstelle des Linux-Kerns angelegt. Der SOLID-Prozess registriert sich beim Kernel, um über diese Ereignisse informiert zu werden. SOLID kann nun über die XML-Schnittstelle von Charon weitere Details zur Beziehung abrufen, wie die Identifikation des anderen IPsec-Gateways. Diese nutzt SOLID, um mit Hilfe des übermittelten Zertifikates die inneren IP-Adressbereiche des korrespondierenden Sicherheits-Gateways sicher zu ermitteln. Für diese werden anschließend Routen angelegt und die IPIP-Tunnel in der iptables-Firewall freigeschaltet. Die Netzwerkgeräte innerhalb der privaten Netze der Sicherheits-Gateways können nun transparent miteinander kommunizieren.

Je nach Konfiguration des SOLID-Gateways werden von der Topologiekontrolle proaktiv weitere Beziehungen zu anderen Gateways aufgebaut, so dass zum Einen eine bestimmte Anzahl Beziehungen bereit gehalten wird und zum Anderen ein strukturiertes Overlay entsteht, in dem SOLID weitere Gateways schnell anhand ihrer inneren IP-Adressen finden kann.

Da nicht immer zu allen SOLID-Gateways eine Sicherheitsbeziehung existieren muss, können Netzwerkgeräte Pakete an Ziele senden, zu denen zunächst keine IPsec-Beziehung besteht. Diese werden von einem Routing-Eintrag erfasst und über eine virtuelle Netzwerkschnittstelle von Linux zum SOLID-Prozess geschickt, wo sie zwischengespeichert werden und reaktiv ein Beziehungsaufbau initiiert wird. Nach dem erfolgreichen Aufbau werden die gepufferten Pakete wieder eingespielt und dem Empfänger zugestellt.

Werden private Adressbereiche innerhalb des VPN verwendet, kann die IPsec-Beziehung unter Umständen nicht sofort eingerichtet werden, da SOLID zunächst die externe IP-Adresse des Sicherheits-Gateways des Zielsystemes ermitteln muss. Dazu setzt der SOLID-Discovery-Algorithmus das bereits erwähnte strukturierte Overlay-Netzwerk ein, welches die internen, privaten IP-Adressen in externe abbilden kann. Bei geschachtelten VPN-Strukturen, also solchen bei denen sich Sicherheits-Gateways hinter anderen Sicherheits-Gateways befinden können, wird gegebenenfalls ein virtueller Pfad zum Ziel-Gateway über andere SOLID-Gateways aufgebaut. Das Weiterleiten von IKE- und ESP-Paketen geschieht durch entsprechende Konfiguration von Firewall und Routing. Nach dem Ermitteln der externen Adresse des Sicherheits-Gatways beziehungsweise nach der Einrichtung des virtuellen Pfades erfolgt der Aufbau der Sicherheits-Beziehung analog zum direkten Aufbau.

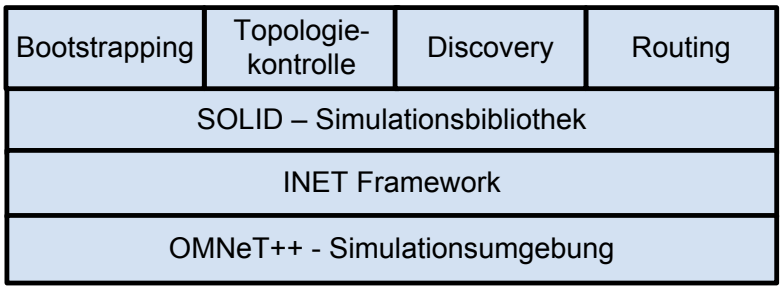

Abbildung 4 – SOLID OMNeT++-Integration

Der Abbau von Sicherheitsbeziehungen erfolgt ebenfalls automatisch. Dieser kann zum einen durch Kommunikationsfehler ausgelöst werden. Zum anderen wird SOLID vom Linux-Kernel über das XFRM-Interface in regelmäßigen Abständen über die Verwendung von Sicherheitsbeziehungen informiert. Werden diese längere Zeit nicht benötigt, veranlasst SOLID einen Abbau.

Eine weitere Besonderheit von SOLID ist das durchgängig ereignisgesteuerte Design, welches eine leichte Integration in den OMNeT++ Simulator [Var01] ermöglicht (siehe **Abbildung 4** und **Abbildung 5**). Dieser kann über eine Bibliothek einkompiliert werden und dient der Entwicklung und dem Test von Bootstrapping-, Discovery-, Routing- und Topologiekontrollmechanismen in großen VPN-Szenarien. Durch die Verwendung des INET Frameworks, können bei den Simulationen auch Eigenschaften der Vermittlungs- und Datensicherungsschicht berücksichtigt werden.

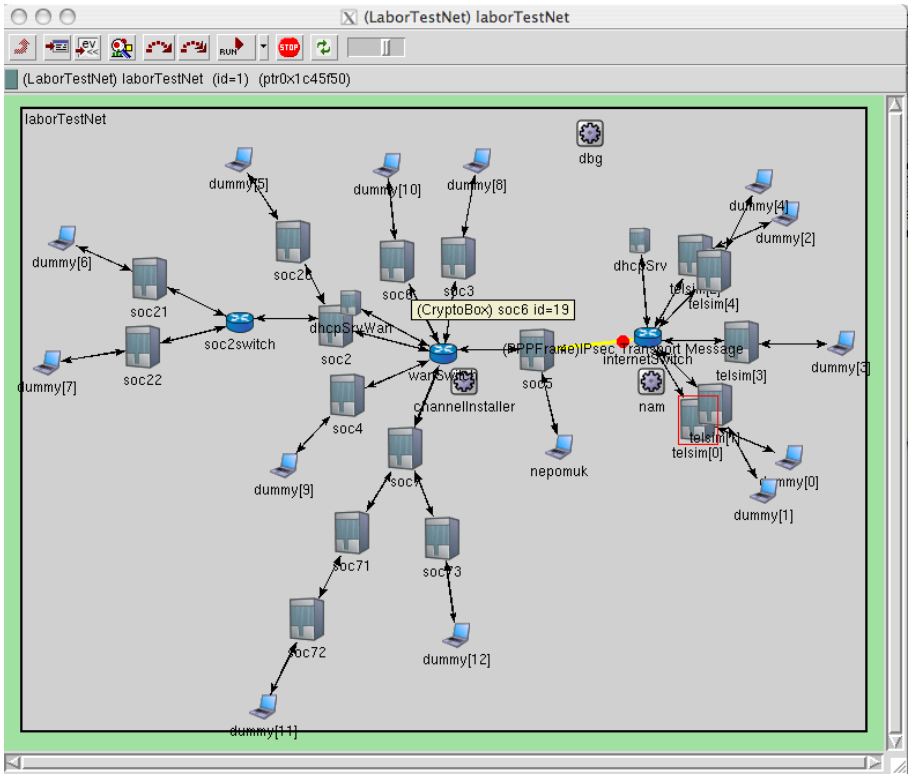

Abbildung 5 – Modelliertes Test-Netzwerk in OMNeT++

Ein weiterer positiver Nebeneffekt des ereignisgesteuerten Designs ist die Verwendung eines einzelnen Threads für Management-Aufgaben, da so Race Conditions in diesem sicherheitskritischen Programm verhindert werden.

5. Bewertung des Ansatzes

Die Bewertung der Softwarearchitektur von SOLID erfolgt anhand der in Abschnitt 2 gegebenen Kriterien.

5.1. Flexibilität

Durch die automatisierte Einrichtung von IPsec-Policies werden Administratoren entlastet. Durch den Verzicht auf manuelle Eingriffe kann das System aber auch schneller auf veränderte VPN-Topologien reagieren. Zusätzlich erlaubt die Entkopplung von Sicherungs- und Routing-Funktionalität, schneller auf Veränderungen im Transportnetzwerk zu reagieren als dies durch die manuelle oder automatisierte Einrichtung von IPsec-Tunneln möglich ist.

Die Komplexität der unterstützten VPN-Topologien kann zudem weitaus komplexer sein, als bei anderen Verfahren. Neben privaten IP-Adressbereichen unterstützt SOLID beispielsweise beliebig geschachtelte IPsec-Gateways und IPsec-Gateways hinter NAT-Routern.

5.2. Reaktives Verhalten

Das temporäre Zwischenspeichern von Paketen erlaubt es SOLID, IPsec-Beziehungen erst bei Bedarf aufzubauen, sodass nicht alle Gateways ständig einen Sicherheitskontext aufrechterhalten müssen. Der Topologiekontrollmechanismus kann parametergesteuert zu einer festgelegten Anzahl von Gateways auch proaktive Verbindungen aufbauen. Nicht-benötigte Sicherheitsbeziehungen werden automatisch abgebaut.

5.3. Sicherheit

Die dynamische Konfiguration von IPsec-Beziehungen birgt zunächst ein Risikopotential, da Änderungen an der VPN-Topologie nicht manuell überprüft werden. Durch die erzwungene Ende-zu-Ende-Verschlüsselung in SOLID ist die Vertraulichkeit, Authentizität und Integrität der Daten allerdings zu jeder Zeit gewährleistet. Externe Angreifer erhalten aufgrund des IPsec-Schutzes der SOLID-Pakete keine neuen Einflussmöglichkeiten auf das VPN. Einzig interne Angreifer könnten prinzipiell mithilfe kompromittierter Sicherheits-Gateways Routing-Angriffe durchführen. Da die Ende-zu-Ende-Verschlüsselung auch interne Angreifer betrifft, werden die Auswirkungen stark reduziert, sodass interne Angreifer, die IPsec-Gateways kontrollieren, lediglich Angriffe auf die Verfügbarkeit ausführen können. Somit wird keine neue Qualität gegenüber Angriffen auf das Routing-System des Transportnetzes erreicht.

5.4. Transparenter Betrieb

Das SOLID-System ist für Endsysteme völlig transparent. Die IPsec-Gateways verhalten sich aus ihrer Sicht wie normale Internet-Router. Allein der reaktive Aufbau von IPsec-Beziehungen kann eine kurze Verzögerung bewirken. Die Dauer der Verzögerung setzt sich in der Regel aus drei Round-Trip-Zeiten für Discovery und IKEv2, sowie der Zeit zur Generierung der IKE-Signaturen zusammen. Insbesondere beim Test von RSA-Zertifikaten auf Smartcards hat sich gezeigt, dass die Generierung der IKE-Signaturen einen signifikanten Einfluss auf die Verzögerung haben kann.

Da Pakete aber während des Beziehungsaufbaus zwischengespeichert und sofort nach erfolgtem Aufbau wiedereingespielt werden, wird die Verzögerung so minimiert, dass sie im Arbeitsablauf nicht zu spüren sind.

5.5. Integrierbarkeit

Ein Ziel bei der Entwicklung des SOLID-Systems war eine Steuerung der verschiedenen Open-Source-Komponenten ohne diese zu modifizieren, und so eine einfache Pflege des Systems zu erreichen. So erfordert SOLID zwar einen aktuellen Linux-Kern, kann diesen aber vollständig unmodifiziert verwenden.

Die Verwendung von IPIP-Tunneln vereinfacht den Einsatz des Charon-Daemons, da Charon zur Zeit noch keine verschachtelten IPsec-Beziehungen unterstützt und eine Erweiterung umfangreiche Änderungen erfordert hätte. Dennoch benötigte er mehrere Patches beispielsweise zur Integration von Smartcards, einige sind jedoch in das strongSwan-Projekt zurückgeflossen und wurden von den Entwicklern in den offiziellen Code integriert. Somit müssen vor dem Einsatz von SOLID an dieser Stelle nur noch zwei Modifikationen vorgenommen werden. Zum einen darf Charon nicht auf Änderungen, die von SOLID am Routing vorgenommen werden, reagieren, sonst würde jede Änderung eine IKE-Reauthentisierung bewirken. Zum anderen müssen die ausgehandelten BEET-Adressen überprüft werden, damit ein kompromittiertes Sicherheits-Gateway nicht die Adresse eines anderen Gateway verwenden kann. Zur Überprüfung werden von SOLID und vom modifizierten Charon IP-Adressbereiche in der IPsec-Zertifikatskette ausgewertet.

5.6. Simulierbarkeit

Komplexe VPN-Szenarien, die Auswirkungen von Ausfällen oder Denial-of-Service-Angriffen, sowie die Reaktionen des Routing-Algorithmus' können ohne weitere Anpassungen vollständig im OMNeT++-Simulator untersucht werden.

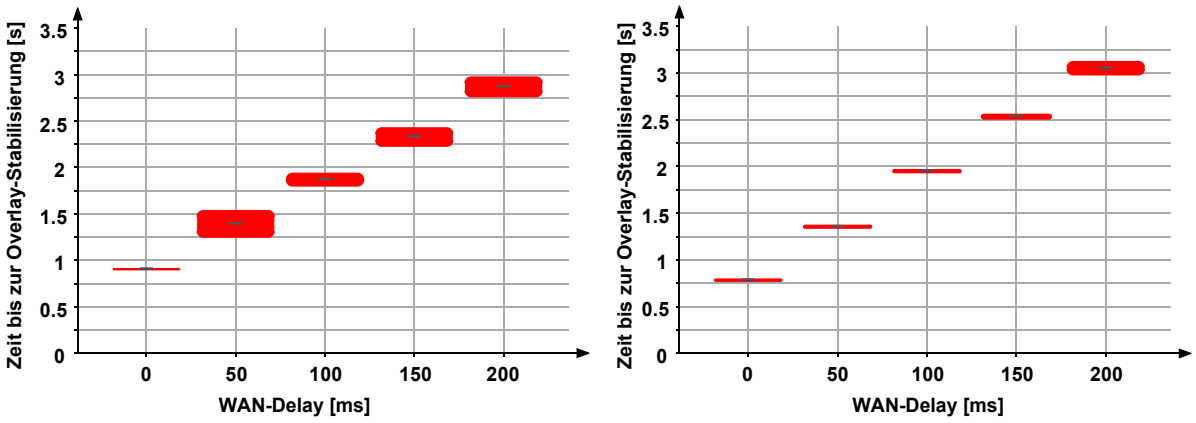

Abbildung 6 - Einfügen eines neuen SOLID-Knotens in Experiment (links) und Simulation (rechts), alle Angaben mit 99% Konfidenzintervall

Ein Beispiel für die Qualität der gewonnen Aussagen ist in Abbildung 6 dargestellt. In mehreren Experimenten wurde der Einfluss der Verzögerung des Transportnetzes auf

die Integration neuer SOLID-Gateways erfasst. Jeder Integrationsvorgang beinhaltet den Aufbau einer Reihe von IPsec-Beziehungen und einen Austausch von Routing-Informationen, sodass mit Abschluss des Vorganges das eingefügte Gateway von allen anderen Gateways des VPN gefunden werden kann. Im untersuchten Szenario wurde ein VPN mit 16 teilweise verschachtelt angeordneten SOLID-Gateways, um ein weiteres Gateway erweitert. Dabei wurde mit einem WAN-Emulator die Verzögerungszeit zwischen 0 und 200 Millisekunden variiert und gleichzeitig ein konstanter Jitter von 10% der Verzögerungszeit hinzugefügt. Die Experimente ergaben einen linearen Anstieg der Einfügedauer von 0,9 Sekunden auf 2,8 Sekunden.

Im OMNeT++-Simulator wurde das gleiche Szenario modelliert, und es ergab sich ein linearer Anstieg von 0,75 Sekunden auf 3,1 Sekunden. Einige Zwischenwerte sind statistisch nicht zu unterscheiden.

Somit können die verwendeten Simulationsmodelle in diesem Beispiel die Wirklichkeit realistisch nachbilden. Desweiteren zeigt das Beispiel, dass selbst ein SOLID-Gateway mit einer großen Netzwerkverzögerung von 200 Millisekunden sich in circa drei Sekunden vollständig in ein VPN integrieren kann. Bei einer Verzögerung von 100 Millisekunden, wie sie beispielsweise bei Transatlantik-Routen auftritt, ist eine noch geringere Integrationszeit von unter zwei Sekunden zu erwarten.

6. Zusammenfassung

Die vorgestellte Softwarearchitektur erlaubt sehr viel dynamischere VPN auf Basis von IPsec als dies bisher der Fall war und kann durch die automatisierte Erstellung von Security Policies den manuellen Konfigurationsaufwand senken. In Linux-basierten VPN-Gateways integriert sich dazu lediglich ein weiterer Hintergrundprozess, und andere Software-Komponenten wie IKE-Daemon und IPsec-Processing müssen nicht oder kaum modifiziert werden, sodass ein einfacherer Zertifizierungsprozess zu erwarten ist. Da die Geräte innerhalb sicherer Netzwerke für den Einsatz von SOLID nicht verändert werden müssen, beschränkt sich ein Rollout auf die Sicherheits-Gateways. Zusätzlich können komplexe Netzwerkszenarien vollständig simulativ evaluiert werden.

7. Ausblick

Die Bestrebungen SOLID in Zukunft weiterzuentwickeln gruppieren sich in zwei Bereiche. Zum einen soll das System zur Marktreife geführt werden, und zum anderen sollen konzeptionelle Weiterentwicklungen die Stärken von SOLID weiterausbauen. In den folgenden Unterabschnitten wird auf diese Aspekte näher eingegangen.

7.1. SINA-Integration

SINA – Secure Inter-Networking Architecture – wurde vom BSI und secunet entwickelt, um klassifizierte Informationen (Verschlusssachen) zu verarbeiten, zu speichern und zu übertragen. Dem Nutzer soll trotz enorm hoher Sicherheitsanforderungen in

vielerlei Hinsicht ein vergleichbares „Look and Feel“ geboten werden, das er aus der offenen Welt kennt.

Die SINA Box als Layer-3-VPN-Gateway basiert im Wesentlichen auf Standard-IPsec-Technologie und kann (beziehungsweise muss) daher wie jedes andere kommerzielle IPsec-fähige Gerät in ein Netzwerkdesign eingeplant werden. Insofern treffen die hier genannten Einschränkungen der bisherigen kommerziellen IPsec-Lösungen gleichzeitig auch für die SINA Box zu. Große behördliche Vernetzungsprojekte mit tausenden von Standorten, aber auch sehr dynamische Einsatzszenarien im militärischen Bereich erfordern umso mehr einfach konfigurierbare, weitestgehend automatisch arbeitende Kryptogeräte, bei denen allerdings ein sehr hoher Sicherheitsanspruch bereits auf konzeptioneller Ebene erhalten bleiben muss. In diesem Umfeld sind auf Grund der hohen Sicherheitsanforderungen keine 'pragmatischen Kompromisse' im Schlüsselmanagement und damit verbundener Zugriffskontrolle haltbar.

SOLID bietet dieses hohe Sicherheitsniveau bei gleichzeitiger vereinfachtem Betrieb standardisierter Netzszenarien und mobiler/dynamischer Konfigurationen. Daher ist SOLID perspektivisch eine sehr sinnvolle Ergänzung der SINA Plattform, die secunet gemeinsam mit der TU Ilmenau weitertreiben möchte.

7.2. Konzeptionelle Weiterentwicklung

Neben weiteren Verbesserungen, um SOLID zur Marktreife zu führen, soll das System auch um konzeptionelle Weiterentwicklungen ergänzt werden. Dazu sind die Entwicklung und Integrationen von Routing-Algorithmen geplant, die beispielsweise robust auf kompromittierte Sicherheits-Gateways reagieren können.

Die Verfügbarkeit von VPN kann weiterhin durch eine bessere Topologiekontrolle erhöht werden, indem SOLID-Gateways verschiedener Sicherheitsstufen sich schalenartig um vitale Kernbereiche des VPN anordnen, und so diese vor Denial-of-Service-Angriffen schützen [BRS09].

Aber auch funktionale Erweiterungen zur Unterstützung von Quality-of-Service und IP-Multicast sind geplant. Letzteres soll mittels Application Layer Multicast (ALM) durch die SOLID-Gateways geschehen, da so eine Unabhängigkeit von der Multicast-Unterstützung des Transportnetzes erreicht wird, und das zu erreichende Sicherheitsniveau wesentlich höher ist.

Da IPsec-Policies in manuell konfigurierten VPN nicht nur für die Topologiekontrolle und für Routing-Entscheidungen eingesetzt werden, sondern auch Zugriffskontrollmechanismen realisieren, soll SOLID auch in dieser Richtung, beispielsweise durch den Einsatz von Capabilities, erweitert werden.

[RSS09] Rossberg, Michael; Strufe, Thorsten; Schaefer, Guenter: Distributed Automatic Configuration of Complex IPsec-Infrastructures, Submitted to DSN 2009.

[Cis05] Bollapragada, Vijay; Khalid, Mohamed; Wainner, Scott: IPSec VPN Design, Cisco Press, 2005.

[Bha08] Y. Bhaiji: Network Security Technologies and Solutions, Cisco Press, 2008.

[SS99] Srisuresh, Pyda; Sanchez, Luis A.: Policy Framework for IP Security, Expired Internet Draft, 1999.

[TE04] Touch, Joe; Eggert, Lars; Wang, Yu-Shun: Use of IPsec Transport Mode for Dynamic Routing, RFC 3884 Proposed Standard 2005.

[NM08] Nikander, Pekka; Melen, Jan: A Bound End-to-End Tunnel (BEET) mode for ESP. IETF Internet Draft, 2008.

[Kau05] Kaufman, Charlie: Internet Key Exchange (IKEv2) Protocol. RFC 4306 Proposed Standard, 2005.

[Var01] Varga, Andras: The OMNeT++ Discrete Event Simulation System. European Simulation Multiconference (ESM), 2001.

[BRS09] Brinkmeier, Michael; Rossberg, Michael; Schaefer, Guenter: Towards a Denial-of-Service Resilient Design of Complex IPsec Overlays. Accepted for IEEE International Conference on Communications (ICC), 2009.