Einsatz von Sicherheitskernen und Trusted Computing

Ahmad-Reza Sadeghi · Marcel Winandy

Horst Görtz Institut für IT-Sicherheit
Ruhr-Universität Bochum
{ahmad.sadeghi, marcel.winandy}@trust.rub.de

Zusammenfassung

Die sichere Übertragung von Daten in offenen Netzen allein reicht nicht aus. Bei der Kommunikation mit anderen Computerplattformen wissen wir oft nicht, in welchem Zustand sich das System am anderen Ende befindet. Völlig neu überarbeitete, sichere Allzweck-Betriebssysteme sind in absehbarer Zeit nicht zu erwarten. Es sind derzeit jedoch zwei Trends zu beobachten: die Einbettung von Sicherheitsfunktionalität in Hardware (Stichwort: Trusted Computing) sowie zunehmender Einsatz von Virtualisierungstechnologie. Die Einbeziehung von Trusted Computing und Virtualisierungstechnologie ermöglicht die Entwicklung von Sicherheitskernen, die sich auf die wichtigsten und grundlegenden Sicherheitsfunktionen beschränken (z.B. sicheres Booten, isolierte Ausführungsumgebungen). Wir skizzieren in dieser Arbeit die Entwicklung moderner Sicherheitskerne mit Trusted Computing am Beispiel einiger Forschungs- und Entwicklungsprojekte und beschreiben mögliche Anwendungsszenarien sowohl für Privatanwender als auch für Anwendungen im Unternehmensumfeld.

1 Einführung

Viele Anwendungen, vor allem im Internet, erfordern eine sichere Verarbeitung und Übertragung von Daten. Zu den sicherheitskritischen Anwendungen gehören beispielsweise Online-Banking, e-Commerce Shops sowie e-Government. Verschlüsselungstechniken und die Digitale Signatur kommen zum Einsatz, um die Datenübertragung kryptographisch abzusichern. Allerdings hat die Erfahrung im e-Commerce-Bereich gezeigt, dass diese Schutzmaßnahmen nicht ausreichend sind. Prominente Angriffe sind Phishing sowie Schadprogramme. Bei Phishing-Angriffen werden Anwender auf gefälschte Webseiten gelockt, um dort ihre Zugangsdaten für z.B. ihr Online-Banking einem Betrüger preiszugeben. Schadprogramme können sich auf dem Rechner des Anwenders einnisten, um dann PIN oder Passwort-Eingaben abzufangen oder Daten und Programme zu verändern.

Die Beispiele zeigen, dass die sichere Übertragung von Daten allein nicht ausreicht. Bei der Kommunikation mit anderen Computerplattformen wissen wir oft nicht, in welchem Zustand sich das System am anderen Ende befindet: Schadeprogramme wie Viren, Würmer und Trojaner könnten vorhanden sein, um Daten abzufangen oder zu manipulieren, wichtige Schutzfunktionen wie Virenscanner könnten außer Kraft gesetzt sein, etc. Vor allem die Komplexität herkömmlicher Betriebssysteme auf PC-Plattformen führt zwar einerseits zu einem (auf jeden Fall gewollten) umfangreichen Funktionsumfang, ist aber andererseits aus gleichem Grund oft auch anfällig für Angriffe von Schadprogrammen. Nicht zuletzt erfordert die sichere Admini-

stration von und der Umgang mit wichtigen Betriebssystemfunktionen ein fundiertes Hintergrundwissen, über welches durchschnittliche PC-Anwender oft nicht verfügen.

Zu warten, bis im Markt völlig neu überarbeitete, sichere Betriebssysteme erscheinen, liefert kurz- und mittelfristig keine Lösung – andere Konzepte müssen her. Das Konzept des Sicherheitskerns (engl. *security kernel*) geht einen anderen Weg als Standardbetriebssysteme: alle Sicherheitsfunktionen werden zentral in einem Kern gebündelt und sind nicht verstreut über die vielen Komponenten des Betriebssystems. Dieser Ansatz ermöglicht eine bessere Evaluierung der Sicherheit, da alle relevanten Funktionen klar definiert sein müssen und keine überflüssige (nicht sicherheitsrelevante) Funktionalität enthalten. Idealerweise ist der Sicherheitskern zudem von der Komplexität her klein genug, um seine Korrektheit formal beweisen zu können. Dieser Vorteil ist zugleich sein größter Nachteil, denn die Entwicklung eines Sicherheitskerns ist entsprechend aufwendig und kostspielig, weshalb dieser Ansatz meist nur bei Anwendungen mit besonders hohen Sicherheitsanforderungen (z.B. im militärischen Bereich) zum Einsatz kommt.

Im IT-Umfeld sind derzeit zwei Trends zu beobachten, die die Entwicklung von Sicherheitskernen unterstützen und vereinfachen können: Einbettung von Sicherheitsfunktionalität in Hardware (Stichwort: Trusted Computing) sowie zunehmender Einsatz von Virtualisierungstechnologie. Ursprünglich getrieben von Einsparungspotentialen bei Energieverbrauch und Betriebskosten, ermöglicht die Virtualisierung nebenbei auch die isolierte Ausführung von verschiedenen Arbeitsumgebungen innerhalb von virtuellen Maschinen auf einem Rechner. So können sicherheitskritische Anwendungen getrennt von anderen Anwendungen in ihren eigenen Betriebssystemumgebungen ausgeführt werden. Die zunehmende Verbreitung von Virtualisierungslösungen, sowohl kommerzieller (z.B. VMware) als auch Open-Source Software (z.B. Xen [DFHH+03], KVM, VirtualBox), und Integration von Virtualisierungsunterstützung durch die CPU-Hersteller (Intel VT [NSLR+06], AMD SVM [AMD05]) macht eine kosteneffiziente Nutzung dieser Technologie möglich. Der zweite Trend, Einbettung von Sicherheitsfunktionen in die Hardware, ist die Basis von *Trusted Computing*: beispielsweise das *Trusted Platform Module* (TPM) ist ein kleiner Zusatzchip nach den Spezifikationen der Trusted Computing Group (TCG) [Gr07], der auf der Hauptplatine bereits vieler PCs und Notebooks zu finden ist und – ähnlich wie eine Smartcard – kryptographische Funktionen sowie geschützten Speicher für kleine Datenmengen bietet. Mit Hilfe des TPMs lässt sich beispielsweise ein sicherer Bootprozess realisieren, d.h. man kann feststellen, in welchem Zustand und mit welcher Softwarekonfiguration der Rechner gestartet wurde. Ergänzt wird dies durch die Integration weiterer Sicherheitsfunktionen direkt in die CPU, z.B. bei Intel TXT [Int08].

Die Einbeziehung von Trusted Computing und Virtualisierungstechnologie ermöglicht die Entwicklung von Sicherheitskernen, die sich auf die wichtigsten und grundlegenden Sicherheitsfunktionen beschränken (z.B. sicheres Booten, isolierte Ausführungsumgebungen) und somit sowohl in absehbarer Zeit als auch mit wirtschaftlich vertretbarem Aufwand realisiert werden können. Diese Sicherheitskerne arbeiten direkt in der Virtualisierungsschicht und bieten sichere Ausführungsumgebungen für Anwendungen, die zusammen mit dem eigentlichen Betriebssystem in virtuellen Maschinen (VMs) ausgeführt werden. Die Umsetzung und Konfiguration feingranularer Sicherheitsmechanismen, welche je nach Anwendungsgebiet sehr komplex und unterschiedlich ausfallen können, wird dabei von den jeweiligen Gastbetriebssystemen in den VMs realisiert. Somit sind nur „grobgranulare“ Sicherheitsfunktionen im Sicherheitskern, was dessen Komplexität und damit Möglichkeit zur Evaluierung überschaubar hält.

Wir beschreiben in diesem Beitrag die Ergebnisse einiger Forschungs- und Entwicklungs-

projekte, die die prototypische Realisierung derartiger Sicherheitskerne als Ziel hatten (Abschnitt 3), und zeigen mögliche Anwendungsszenarien hierfür in unterschiedlichen Bereichen auf, von privaten Anwendungen bis hin zum Unternehmenseinsatz (Abschnitt 4). Wir beginnen zunächst mit der Zusammenfassung einiger Grundlagen zu Trusted Computing im folgenden Abschnitt.

2 Grundlagen zu Trusted Computing

Das Herzstück von Trusted Computing nach den TCG Spezifikationen ist das TPM [Gr07]. Das TPM bietet kryptographische Funktionen (u.a. asymmetrische Schlüsselgenerierung, Verschlüsselung und Signatur, Hashwertberechnung, Zufallszahlengenerator) und geschützten Speicher für kleine Datenmengen (in erster Linie für kryptographische Schlüssel und monotone Zähler). Die beiden Hauptschlüssel des TPMs sind zum Einen der *Endorsement Key*, der die eindeutige Identität des TPM-Chips repräsentiert, und der *Storage Root Key* (SRK), mit dem andere vom TPM generierte (aber außerhalb des TPM gespeicherte) Schlüssel verschlüsselt und entschlüsselt werden. Die privaten Schlüsselanteile geschützter Schlüssel verlassen dabei niemals das TPM.

Beim Hochfahren des Rechners werden Prüfwerte der geladenen Software (die sogenannte Plattformkonfiguration) in den geschützten Bereich des TPM, die *Platform Configuration Registers* (PCRs) geschrieben. Dabei berechnet jede Komponente im Boot-Prozess zuerst den Prüfwert der jeweils nächsten und übergibt dann erst die Kontrolle. Da in den PCRs immer nur ein kumulativer Hashwert der jeweils zuvor gespeicherten Werte geschrieben werden kann und die kryptographische Einweg-Hash-Funktion SHA-1 zum Einsatz kommt, ist die Reihenfolge der Boot-Komponenten anhand der Werte in den PCRs überprüfbar.

Das TPM bietet die Möglichkeit, die Hashwerte der PCRs mit bestimmten (vom TPM geschützten) Schlüsseln zu verknüpfen. Daraus resultieren zwei wichtige Verfahren:

- *Attestation* erlaubt die kryptographisch abgesicherte Überprüfung der Plattformkonfiguration von einem entfernten Rechner aus. Dabei signiert das TPM seine aktuellen PCR-Werte mit einem *Attestation Identity Key* (AIK), welches die Rolle eines Pseudonyms für die Identität des TPMs spielt. Die Echtheit des AIKs wird entweder durch ein Zertifikat bescheinigt, das das TPM mit seinen Endorsement Key über eine Zertifizierungsstelle (sogenannte *Privacy-CA*) erhält, oder über ein spezielles kryptographisches Protokoll (*Direct Anonymous Attestation* [BrCC04]). Eine verifizierende Partei kann dann anhand der signierten PCR-Werte und eines Protokolls des Bootvorgangs überprüfen, welche Komponenten beim Hochfahren der Plattform geladen wurden.
- *Sealing* kann den Zugriff von (verschlüsselten) Daten an die Plattformkonfiguration des Rechners binden, d.h. die Daten (genauer der zugehörige Entschlüsselungsschlüssel) können nur dann wieder entschlüsselt werden, wenn die gleichen PCR-Werte im TPM vorhanden sind wie zum Zeitpunkt des Verschlüsselns. Dadurch kann beispielsweise verhindert werden, dass sensitive Daten in einer nachträglich manipulierten Betriebssystemumgebung entschlüsselt werden.

Aufbauend auf diesen Grundfunktionen können eine Vielzahl weiterer Sicherheitsfunktionalitäten realisiert werden. Beispiele sind sicheres bzw. authentifiziertes Booten [KSS07], erweiterte Integritätsprüfungen im Betriebssystem [SZJv04], verschlüsselte und die Integrität der Endpunkte einbeziehende Kommunikationskanäle (sogenannte *Trusted Channels*) [GoPS06,

AGSS+08]. Der Einsatz eines TPMs alleine genügt jedoch nicht, da es sich um einen passiven Chip handelt. Man braucht noch eine entsprechende Softwarebasis, die das TPM geeignet und effizient nutzt; mit anderen Worten: ein Trusted-Computing-fähiges Betriebssystem.

3 Entwicklung moderner Sicherheitskerne mit Trusted Computing

Ziel ist die Entwicklung eines Sicherheitskerns, der verschiedene Anwendungsbereiche auf einem Rechner isoliert voneinander ausführen kann und die Funktionen des TPMs nutzt, um die Bereiche zu schützen und gegenüber Anwendern und anderen Rechnern im Netzwerk zu attestieren. Neuartig gegenüber früheren Ansätzen von Sicherheitskernen ist die Beschränkung auf grundlegende Sicherheitsfunktionen, die grobgranulare Sicherheitseigenschaften realisieren können. Zentrale Idee hierbei ist die Verwendung von Virtualisierungstechnologie und das Überlassen von feingranularen, anwendungsspezifischen Sicherheitseigenschaften den jeweiligen Gastbetriebssystemen, die in virtuellen Maschinen ausgeführt werden. Dies ermöglicht die Konzentration auf wenige Sicherheitsmechanismen im Sicherheitskern und bedeutet geringere Komplexität, deren Korrektheit es zu verifizieren gilt. Die Realisierung des Sicherheitskerns innerhalb der Virtualisierungsschicht erlaubt zugleich die Wiederverwendung von bisherigen Applikationen, was für eine Akzeptanz im Markt besonders wichtig ist. Im Folgenden stellen wir zwei Ansätze für derartige Sicherheitskerne vor: ein Mikrokern-basierter Ansatz, bei dem ein Mischbetrieb von nativen Mikrokern-Anwendungen und virtuellen Maschinen möglich ist, und ein rein Hypervisor-basierter Ansatz, der nur mit virtuellen Maschinen arbeitet.

3.1 Mikrokern-basierter Ansatz

Turaya ist ein Mikrokern-basierter Sicherheitskern, der aus einem vom Deutschen Bundesministerium für Wirtschaft und Arbeit geförderten Projekt [EMS] hervorgegangen und als Open Source Software verfügbar ist. Basis von Turaya ist der L4 Mikrokern [Lied95]. Ein Mikrokern-Betriebssystem hat gegenüber anderen Betriebssystemen einen sehr kleinen Kernel, der allein im privilegierten Modus des Prozessors ausgeführt wird. Alle anderen Programme werden im nicht-privilegierten Modus ausgeführt, darunter fallen auch alle Gerätetreiber und Basisfunktionen eines Betriebssystems selbst wie z.B. Speicherverwaltung. Dies hat den Vorteil, dass der Kernel durch fehlerhafte Prozesse oder Schadprogramme in Treibern nicht korrumpiert werden kann. Prozesse kommunizieren miteinander durch Austausch von Nachrichten, und der Mikrokern kontrolliert dieses Nachrichtenaustausch. Heutige Mikrokerne wie der L4 können dies hinreichend effizient erledigen. Allerdings erfordert der Entwurf der Kommunikationsschnittstellen der Prozesse im Vergleich zu monolithischen Kernelarchitekturen mehr Sorgfalt und Aufwand. Dies kann sich jedoch bei der Evaluierung des Sicherheitskerns auf Grund eines sauberen Designs wieder auszahlen, was den Nachteil in einen weiteren Vorteil umkehrt.

Basierend auf dem L4 Mikrokern besteht Turaya vor allem aus zwei Schichten von Diensten: *Resource Management Layer* und *Trusted Software Layer*. Die Hauptaufgabe der Resource Management Layer besteht in der Bereitstellung abstrakter Schnittstellen zu darunter liegenden Hardwareressourcen wie Interrupts, Arbeitsspeicher und anderer Hardware. Diese Ebene verteilt die zur Verfügung stehenden Ressourcen und realisiert die Zugriffskontrolle auf die Objekttypen, die dieser Ebene bekannt sind. Die Trusted Software Layer enthält Dienste, die die Schnittstellen der darunter liegenden Dienste durch Sicherheitsmechanismen und Isolierung der Anwendungen erweitert, welche über diese Ebene eingesetzt werden. Dabei werden zur Rea-

lisierung der Sicherheitsmechanismen die Trusted Computing Funktionalitäten der Hardwareschicht eingesetzt. Auf dem L4-basierten Turaya können sowohl virtuelle Maschinen (derzeit L4Linux – ein paravirtualisiertes Linux) als auch direkt native Prozesse als Anwendungen ausgeführt werden. Abbildung 1 zeigt die Architektur von Turaya.

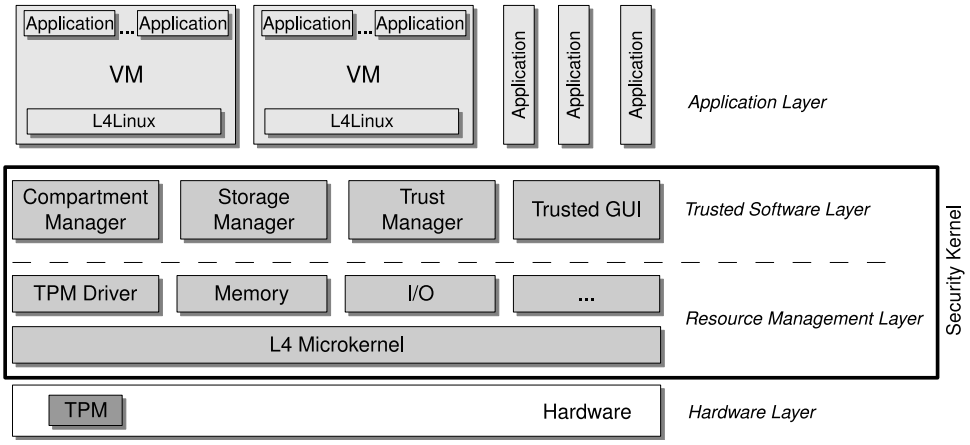

Abb. 1: Architektur von Turaya auf L4

Zu den wichtigsten Sicherheitsdiensten von Turaya gehören:

- *Secure Booting*: Starten der Rechnerplattform in einen authentifizierten Zustand. Der Bootloader berechnet die Integritätsprüfwerte der Komponenten des Sicherheitskerns und überträgt sie in die PCRs des TPM.
- *Storage Manager*: Bereitstellung von sicherem Speicher, der mit von der Anwendung zu bestimmenden Sicherheitseigenschaften versehen werden kann. Dazu gehört die Anwendung von Sealing, um Daten die die Plattformkonfiguration zu binden, sowie die monotonen Zähler des TPMs, um die Aktualität der gespeicherten Daten zu gewährleisten (schützt vor *Replay*-Angriffen).
- *Trust Manager*: Bereitstellung von Trusted Channels, Zertifikaten und Durchführung von Attestation.
- *Compartment Manager*: Isolierung von Anwendung in *Compartments*, Berechnung und Bereitstellung von Integritätsprüfwerten von Compartments, Durchsetzung von Zugriffsrechten von Compartments auf Ressourcen oder Kommunikation mit anderen Compartments.
- *Trusted GUI*: Graphisches Benutzeroberflächensystem, das die Authentifizierung von Anwendungen gegenüber dem Benutzer ermöglicht (sog. *trusted path*) und die Ein-/Ausgabekanäle des Benutzers zwischen einzelnen Compartments isoliert und somit schützt.

3.2 Hypervisor-basierter Ansatz

OpenTC [Ope] ist ein von der EU gefördertes Projekt, welches die Entwicklung eines Trusted Computing Frameworks für sichere Betriebssysteme basierend auf Virtualisierungstechnologie als Ziel hat [KLRS+06]. Das Trusted Computing Framework läuft sowohl mit einem L4-Mikrokern-basiertem Virtualisierungssystem als auch mit einem reinen Hypervisor-basierten Ansatz. Als Hypervisor kommt die Open-Source Software Xen [DFHH+03] zum Einsatz.

Xen ist eigentlich auf die reine Virtualisierung spezialisiert, enthält aber bereits die Sicherheitsarchitektur sHype [SJVP+05], die ein verbindliches Zugriffskontrollmodell für virtuelle Maschinen auf virtuelle Ressourcen umsetzen kann. Dadurch lässt sich sowohl die Isolierung von VMs als auch die kontrollierte Kommunikation durch gemeinsame Ressourcennutzung von VMs realisieren. Innerhalb des Projektes OpenTC wurde ein Sicherheitskern basierend auf Xen entwickelt, der zusätzlich Trusted Computing Funktionalitäten nutzt, um weitere Sicherheitsmechanismen, wie z.B. Trusted Channels, zu implementieren.

Der Sicherheitskern-Prototyp von OpenTC umfasst neben dem Xen Hypervisor auch eine dedizierte virtuelle Maschine, die *Dom0*. Diese VM hat als einzige direkten Hardwarezugriff und enthält die Gerätetreiber, die nicht bereits vom Xen Hypervisor abgedeckt werden. Andere VMs müssen über die Dom0 mit diesen Geräten kommunizieren. Die Dom0 ist damit ein Intermediär und Teil des Sicherheitskerns. Zusätzlich zu den Gerätetreibern enthält die Dom0 auch die Programme zum Management von Compartments (VMs), sicheren Datenspeichern (Sealing) und Implementierung von Trusted Channels. Abbildung 2 zeigt die grobe Architektur des OpenTC Sicherheitskerns auf Xen.

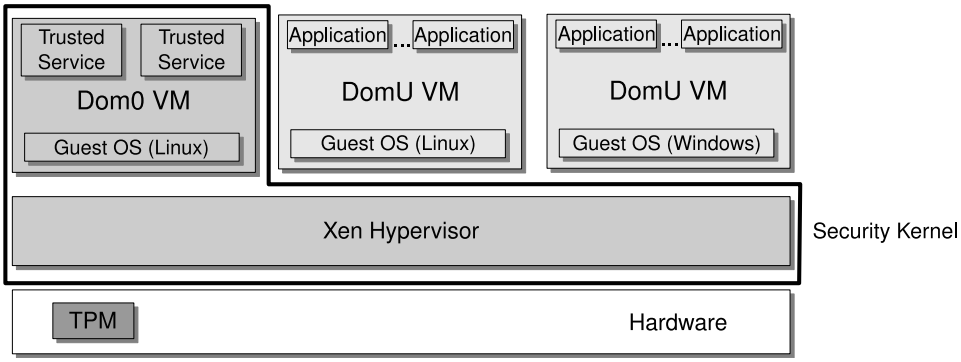

Abb. 2: Architektur von OpenTC auf Xen

Dom0 enthält als Teil des Sicherheitskerns zwar wiederum ein eigenes Betriebssystem und erhöht die Komplexität, dies lässt sich jedoch auf zweierlei Wegen wieder entgegenwirken. Zum einen hat die VM von Dom0 eine dedizierte Aufgabe und verlangt generell kein Allzweck-Betriebssystem mit allen seinen verschiedenen Anwendungen und Programmen. Das System in Dom0 kann ein speziell für diesen Zweck, stark an Funktionalität reduzierter Betriebssystemkern mit genau definierten Programmen sein. Zudem wird der Kernel des Gastbetriebssystems von Dom0 nicht im privilegierten Prozessormodus ausgeführt (dies ist allein dem Xen Hypervisor vorbehalten). Der zweite Weg zur Komplexitätsreduzierung folgt dem Ansatz des Mikrokern-basierten Sicherheitskerns und versucht die Dom0 in viele kleine Teile zu zerlegen [AnMD07]. Jede Komponente läuft dann mit einer Art Mini-Betriebssystem, welches nur Basisfunktionalitäten enthält, um den jeweiligen Trägerprozess ausführen zu können.

Neben OpenTC ist Terra [GPCR+03] ein weiterer Hypervisor-basierter Sicherheitskern, der Trusted Computing Funktionalität nutzt und integriert.

4 Anwendungsszenarien

In diesem Abschnitt beschreiben wir einige Anwendungsszenarien, die mit den oben beschriebenen Sicherheitskernen realisiert werden können. Die gezeigten Ergebnisse beruhen auf Pro-

totypen, die zum Teil während und innerhalb der genannten Projekte entwickelt wurden.

4.1 Sicheres Online-Banking

Ein zentraler Sicherheitsaspekt beim Online-Banking (wie auch allgemein im e-Commerce) ist die Authentifizierung des Benutzers beim Server des Diensteanbieters. Hierbei ist die Verwendung von Benutzername und Passwort/PIN die gängige Methode. Ein Schwachpunkt dieses Verfahrens ist, dass wenn ein Angreifer das Passwort „abhört“, er sich mit diesen Daten als der legitime Benutzer beim Bank-Server ausgeben kann und Transaktionen in dessen Namen ausführt. Dieser Angriff wird Phishing (aus engl. *password fishing*) genannt. Neben Phishing-Angriffen über gefälschte Webserver, die vorgeben die Originalseite zu sein, findet eine andere Klasse von Phishing-Angriffen, die Schadprogramme wie z.B. Keylogger auf dem lokalen Rechner des Anwenders ausnutzen, immer weitere Verbreitung [CNE06, PMMW+07].

Um sich gegen Phishing-Angriffen sowohl über gefälschte Webserver als auch durch lokale Schadprogramme zu schützen, hilft ein Wallet als Authentifizierungsagent des Anwenders [GSSW07]. Das Wallet ist wie eine Brieftasche für Passwörter. Der Anwender speichert seine Login-Daten fürs Online-Banking sowie andere Webserver einmalig im Wallet, das Wallet meldet den Anwender automatisch an, wenn der Anwender auf eine entsprechende Webseite geht. Das Wallet übernimmt dabei die Prüfung der Legitimität des Webservers anhand von SSL-Zertifikaten und trägt die Login-Daten ohne Zutun des Anwenders ein. Das Wallet kann sogar die Passwörter des Anwenders durch „starke“ Passwörter ersetzen, d.h. solche mit hoher Entropie, und sie vor dem Anwender verbergen, damit er sie nicht aus Versehen auf andere Art und Weise einer betrügerischen Webseite verraten kann. Um das Wallet und dessen Benutzung durch Ausspähen von Schadprogrammen zu schützen, die sich unter Umständen auch im Browser befinden können, wird das Wallet in einer vom Browser (und dessen Betriebssystem) getrennten Ausführungsumgebung ausgeführt. Dazu wird der Sicherheitskern benötigt, der Wallet und Browser samt andere Anwendungen in zwei unterschiedliche Compartments ausführt. Abbildung 3 zeigt die Architektur des Wallets.

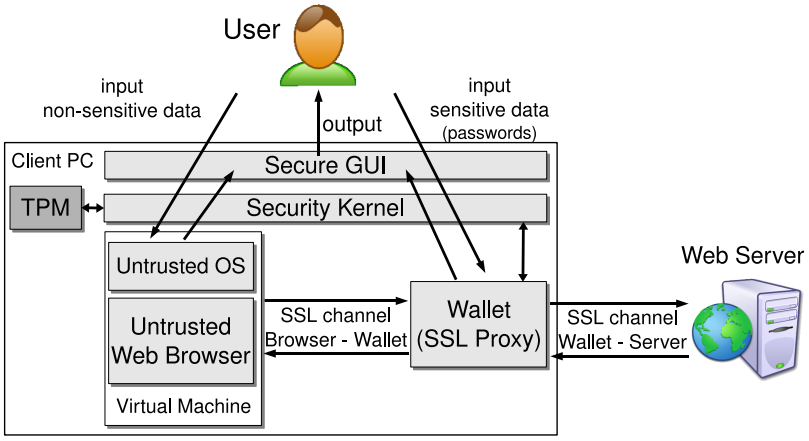

Abb. 3: Wallet-basierte Lösung zum sicheren Login auf Webseiten

Das Wallet agiert als Web-Proxy zwischen dem Browser-Compartment und dem eigentlichen Web-Server, d.h. es baut seine eigene SSL-geschützte Verbindung zum Web-Server auf und kann den Datenverkehr zum Browser überwachen bzw. auch modifizieren. Wenn das Wallet beispielsweise ein Login-Formular erkennt, dann werden die Eingabefelder für den Browser

ausgeblendet und stattdessen die Login-Daten direkt eingefügt, wenn der Benutzer auf dem Anmeldeknopf im Webbrowser drückt. So gelangen die Passwörter niemals in das Browser-Compartment, sondern werden direkt vom Wallet zum Web-Server geschickt, sofern dieser der legitime Server ist, zu dem das Passwort ursprünglich zugeordnet wurde. Wenn das Client-System heruntergefahren wird, werden die Passwörter, die im Wallet gespeichert sind, an die Plattformkonfiguration des Sicherheitskerns und des Wallets mittels Sealing geschützt. Ein Prototyp des Wallets wurde auf der Turaya-Plattform entwickelt, Details sind in [GSSW07] zu finden. Eine ähnliche Anwendung von Virtualisierung zum Schutz von Anwenderdaten liefert [KwDu07].

4.2 Sicherer Zugang zum Unternehmensnetzwerk

Um einen sicheren Zugang zum Unternehmensnetzwerk über das Internet zu ermöglichen, wird üblicherweise ein Virtuelles Privates Netzwerk (VPN) realisiert, indem der Netzwerkverkehr zwischen dem Rechner des Mitarbeiters und einem Gateway zum internen Unternehmensnetzwerk verschlüsselt wird, so dass andere Netzwerkknoten im Internet die Kommunikation nicht „belauschen“ können. Gängige Produkte sind VPN-Client Software, die auf dem Standardbetriebssystem des Mitarbeiter-Rechners laufen. Die kryptographischen Schlüssel für die VPN-Verbindung werden in Dateien oder im Arbeitsspeicher gespeichert, auf den das Betriebssystem natürlich vollständig Zugriff hat. Wenn auf dem Client-Rechner jedoch Schadprogramme laufen, die sich Systemprivilegien holen können (z.B. ein Trojaner in einem Gerätetreiber oder ein Rootkit), dann können die geheimen Authentifikationsdaten für die VPN-Verbindung ausgelesen und einem Angreifer leicht übermittelt werden. Aufgrund der Komplexität von Standardbetriebssystemen und der großen Vielfalt von Trojanern können Schutzprogramme wie Antivirensoftware nur bedingt gegen derartige Angriffe helfen.

Analog zu der Wallet-Architektur kann die Ver- und Entschlüsselung von Netzwerkpaketen in einem isolierten VPN-Client geschehen und alle anderen Anwendungen des Benutzers laufen mit dem Standardbetriebssystem in einer virtuellen Maschine. Der Sicherheitskern sorgt wieder dafür, dass diese beiden Compartments isoliert voneinander laufen und nur der Netzwerkverkehr durchgeleitet wird. Die kryptographischen Schlüssel des VPNs sind für die virtuelle Maschine und dessen Gastbetriebssystem nicht erreichbar und somit vor Ausspähen durch Schadprogramme geschützt. Die persistente Speicherung der VPN-Zertifikate wird wiederum durch die Sealing-Funktion des TPMs an die Plattformkonfiguration des Sicherheitskerns und des VPN-Clients gebunden. Zusätzlich kann das VPN-Gateway mit Hilfe der Attestation-Funktion eine Überprüfung der Plattformkonfiguration des Clients durchführen und so feststellen, ob der Mitarbeiter mit einer gültigen Ausführungsumgebung arbeitet, und ansonsten den Zugang zum VPN verweigern. Ein Prototyp wurde ebenfalls auf der Turaya-Plattform realisiert [ASSS+06].

Neben eines VPN-Zugangs zum Unternehmensnetzwerk von außen, kann auch der Zugang intern durch Trusted Computing Mechanismen kontrolliert werden. Die Grundidee ist die, dass ein Server oder Switch mittels Attestation die Plattformkonfiguration der Client-Rechner, die sich verbinden wollen, überprüfen kann. So werden im Netzwerk nur solche Client-Rechner verbunden, die über eine von der Netzwerk-Policy erlaubten Konfiguration verfügen. Die TCG hat hierzu eine eigene Spezifikation „Trusted Network Connect“ (TNC) [Gr08], und es gibt ein Forschungsprojekt [tNA], welches eine offene Netzwerkzugriffskontroll-Lösung basierend auf TNC entwickelt.

4.3 Verschlüsselung von Festplatten und mobilen Datenträgern

Daten auf mobilen Computern (Laptops, Notebooks) und bei mobilen Datenträgern (USB-Sticks) müssen besonders geschützt werden, denn wenn die Geräte verloren gehen, sind auch die Daten verloren bzw. können in die Hände Unbefugter gelangen. Die Verschlüsselung von Festplatten und mobilen Datenträgern ist daher oft eine wichtige Anforderung. Entsprechende Produkte finden sich in einer Vielzahl am Markt. Jedoch ist auch hier oft der Schutz der kryptographischen Schlüssel einem Standardbetriebssystem überlassen. Windows Vista nutzt in seiner Festplattenverschlüsselung BitLocker [Micr05] hingegen (optional) bereits das TPM zum Schutz der Schlüssel.

Die Anwendung von TPM und Sicherheitskern für Festplattenverschlüsselung wurde prototypisch ebenfalls in [ASSS+06] gezeigt. Die Verschlüsselungskomponente läuft in einer geschützter Ausführungsumgebung, analog wie bei der VPN-Lösung. Das Konzept lässt sich dahingehend erweitern, dass eine automatische und transparente Verschlüsselung von mobilen Datenträgern (USB-Sticks) jederzeit erfolgt. Abbildung 4 zeigt den schematischen Aufbau einer solchen transparenten Datenträgerverschlüsselung für virtuelle Maschinen.

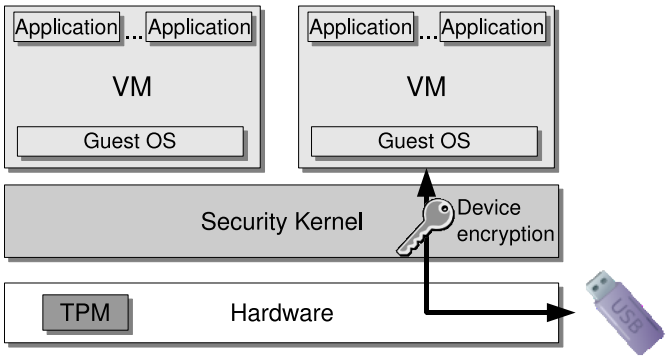

Abb. 4: Transparente Verschlüsselung von mobilen Datenträgern

Die Storage Management Komponente im Sicherheitskern übernimmt die automatische Verschlüsselung der Daten. Virtuelle Maschinen (Compartments) des Anwenders und dessen Gastbetriebssysteme bekommen nur virtuelle Geräte zu sehen, sie haben nie direkten Hardware-Zugriff auf Festplatte oder USB-Stick. So kann man den kryptographischen Schlüssel schützen und gleichzeitig dafür sorgen, dass mobile Datenträger immer verschlüsselt werden. Die Schlüssel selbst können wieder mit Hilfe des TPMs an die Plattformkonfiguration des Client-Rechners gebunden werden und/oder in Kombination mit Attestation des Clients von einem zentralen Schlüsselmanagement im Unternehmensnetz verteilt werden.

4.4 Trusted Virtual Domains - geschützte virtuelle Netze

Bringt man die Verwendung von virtuellen Maschinen, sicheren virtuellen Netzen und Datenträgersicherheit auf einen höheren Abstraktionslevel, gelangt man zum Konzept der *Trusted Virtual Domain* (TVD) [BGJJ+05]. TVDs sind ein neues Konzept zur Verwaltung von isolierten virtuellen Netzwerken. Im Unterschied zu VPNs handelt es sich dabei nicht um eine spezielle Technik, sondern um ein abstraktes Modell. Zur Realisierung von TVDs können VPNs eingesetzt werden, aber es sind auch andere Mechanismen zusätzlich notwendig. Entscheidender

Unterschied zu reinen VPNs ist der, dass alle Netzwerkknoten innerhalb einer TVD die Netzwerkknoten anderer TVDs nicht ßehen"können. Es können somit mehrere virtuelle Netze isoliert voneinander konfiguriert werden. Zudem ist es durch Virtualisierungstechnologie möglich, mehrere Knoten unterschiedlicher TVDs auf einer physikalischen Plattform laufen zu lassen, entsprechende Sicherheitsmechanismen in der Virtualisierungsschicht zu Isolierung der virtuellen Maschinen vorausgesetzt. Sicherheitskerne mit Trusted Computing und Virtualisierungs-Support sind somit geeignete Grundbausteine für eine TVD-Infrastruktur.

Anwendungen für TVDs finden sich in virtuellen Rechenzentren [BCPS+08], wo die Rechenprozesse und Daten verschiedener Kunden oder Abteilungen in jeweils isolierte TVDs eingeteilt werden, aber dennoch je nach Anforderung und Verfügbarkeit von Rechenkapazität, die einzelnen Rechenknoten innerhalb von virtuellen Maschinen verteilt und verwaltet werden. Dies ermöglicht bei vergleichbarer Sicherheit eine höhere Flexibilität, da virtuelle Maschinen beispielsweise auf andere Rechnerhardware im laufenden Betrieb verschoben werden können, wenn dort Kapazität frei wird. Ein weiteres Anwendungsszenario für TVDs ist Enterprise-Rights-Management im Unternehmen [GSSU+08].

5 Fazit

Da völlig neu entwickelte sichere Allzweck-Betriebssysteme noch für geraume Zeit auf sich warten lassen, ist die Verwendung von Virtualisierungstechnologie und Trusted Computing in einem auf die grundlegenden Sicherheitsfunktionen beschränkten Sicherheitskern ein vielversprechender Ansatz. Weitergehende und feingranulare Sicherheitsfunktionen können separat in den höheren Schichten (Gast-Betriebssysteme in virtuellen Maschinen) anwendungsspezifisch hinzugefügt werden. Die Anwendungsszenarien für derartige Sicherheitskerne sind vielfältig, reichen von e-Commerce Anwendungen für private Nutzer bis hin zu Lösungen im Unternehmenseinsatz. Erste Prototypen dieser Sicherheitskerne wurden in verschiedenen Forschung- und Entwicklungsprojekten bereits realisiert und mit Anwendungsbeispielen demonstriert. Durch die Beteiligung von Industriepartnern ist davon auszugehen, dass die Ergebnisse dieser Projekte in zukünftige Produktentwicklungen einfließen werden. Die zunehmende Unterstützung von Virtualisierung und Trusted Computing Funktionen seitens der Hardware-Hersteller (z.B. Intel TXT) stützt diese Prognose.

Literatur

[AGSS+08] F. Armknecht, Y. Gasmi, A. R. Sadeghi, P. Stewin, M. Unger, G. Ramunno, D. Vernizzi: An efficient implementation of trusted channels based on openssl. *In: STC '08: Proceedings of the 3rd ACM workshop on Scalable trusted computing*, ACM, New York, NY, USA (2008), 41–50.

[AMD05] AMD: AMD64 Virtualization Codenamed "Pacifica" Technology — Secure Virtual Machine Architecture Reference Manual. Tech. Rep. Publication Number 33047, Revision 3.01, AMD (2005).

[AnMD07] M. J. Anderson, M. Moffie, C. I. Dalton: Towards Trustworthy Virtualisation Environments: Xen Library OS Security Service Infrastructure. Tech. Rep. HPL-2007-69, Hewlett-Packard Laboratories (2007).

[ASSS+06] A. Alkassar, M. Scheibel, C. Stüble, A.-R. Sadeghi, M. Winandy: Security Ar-

chitecture for Device Encryption and VPN. *In: Proceedings of Information Security Solutions Europe (ISSE 2006)*, Vieweg-Verlag (2006), 54–63.

[BCPS+08] S. Berger, R. Cáceres, D. Pendarakis, R. Sailer, E. Valdez, R. Perez, W. Schildhauer, D. Srinivasan: TVDc: managing security in the trusted virtual datacenter. In: *SIGOPS Oper. Syst. Rev.*, 42, 1 (2008), 40–47.

[BGJJ+05] A. Bussani, J. L. Griffin, B. Jasen, K. Julisch, G. Karjoth, H. Maruyama, M. Nakamura, R. Perez, M. Schunter, A. Tanner, L. V. Doorn, E. V. Herreweghen, M. Waidner, S. Yoshihama: Trusted Virtual Domains: Secure Foundations for Business and IT Services. Tech. Rep. Research Report RC23792 (2005).

[BrCC04] E. Brickell, J. Camenisch, L. Chen: Direct Anonymous Attestation. *In: Proceedings of the 11th ACM Conference on Computer and Communications Security* (2004), 132–145.

[CNE06] New Trojans plunder bank accounts (2006), `http://news.cnet.com/2100-7349_3-6041173.html`.

[DFHH+03] B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, I. Pratt, A. Warfield, P. Barham, R. Neugebauer: Xen and the Art of Virtualization. *In: Proceedings of the ACM Symposium on Operating Systems Principles*, ACM (2003), 164–177.

[EMS] European Multilaterally Secure Computing Base (EMSCB). `http://www.emscb.org`.

[GoPS06] K. Goldman, R. Perez, R. Sailer: Linking Remote Attestation to Secure Tunnel Endpoints. *In: STC '06: Proceedings of the First ACM Workshop on Scalable Trusted Computing* (2006), 21–24.

[GPCR+03] T. Garfinkel, B. Pfaff, J. Chow, M. Rosenblum, D. Boneh: Terra: A Virtual Machine-based Platform for Trusted Computing. *In: 19th ACM Symposium on Operating Systems Principles (SOSP'03)*, ACM (2003), 193–206.

[Gr07] Trusted Computing Group: TPM Main Specification. Specification Version 1.2 rev. 103 (2007), `https://www.trustedcomputinggroup.org/specs/TPM/`.

[Gr08] Trusted Computing Group: Trusted Network Connect. Specification Version 1.3 (2008), `https://www.trustedcomputinggroup.org/specs/TNC/`.

[GSSU+08] Y. Gasmi, A.-R. Sadeghi, P. Stewin, M. Unger, M. Winandy, R. Husseiki, C. Stüble: Flexible and secure enterprise rights management based on trusted virtual domains. *In: STC '08: Proceedings of the 3rd ACM workshop on Scalable trusted computing*, ACM, New York, NY, USA (2008), 71–80.

[GSSW07] S. Gajek, A.-R. Sadeghi, C. Stüble, M. Winandy: Compartmented Security for Browsers – Or How to Thwart a Phisher with Trusted Computing. *In: 2nd Intl. Conference on Availability, Reliability and Security (ARES 2007)* (2007), 120–127.

[Int08] Intel Trusted Execution Technology Software Development Guide. Tech. Rep. Document Number: 315168-005, Intel Corporation (2008).

[KLRS+06] D. Kuhlmann, R. Landfermann, H. V. Ramasamy, M. Schunter, G. Ramunno, D. Vernizzi: An Open Trusted Computing Architecture – Secure Virtual Machines Enabling User-Defined Policy Enforcement. Tech. Rep. RZ 3655 (#99675), IBM Research (2006).

[KSS07] U. Kühn, M. Selhorst, C. Stüble: Realizing Property-Based Attestation and Sealing with Commonly Available Hard- and Software. *In: STC '07: Proceedings of the 2nd ACM Workshop on Scalable Trusted Computing*, ACM Press (2007), 50–57.

[KwDu07] P. C. S. Kwan, G. Durfee: Practical Uses of Virtual Machines for Protection of Sensitive User Data. *In: Information Security Practice and Experience Conference (ISPEC 2007)*, Springer (2007), 145–161.

[Lied95] J. Liedtke: On micro-kernel construction. *In: SOSP '95: Proceedings of the 15th ACM Symposium on Operating System Principles*, ACM (1995), 237–250.

[Micr05] Microsoft: Secure Startup - Full Volume Encryption: Technical Overview. `http://www.microsoft.com/whdc/system/platform/pcdesign/secure-start_tech.mspx` (2005).

[NSLR+06] G. Neiger, A. Santoni, F. Leung, D. Rodgers, R. Uhlig: Intel Virtualization Technology: Hardware Support for Efficient Processor Virtualization. In: *Intel Technology Journal*, 10, 3 (2006).

[Ope] Open Trusted Computing. `http://www.opentc.net`.

[PMMW+07] N. Provos, D. McNamee, P. Mavrommatis, K. Wang, N. Modadugu: The ghost in the browser analysis of web-based malware. *In: HotBots'07: Proceedings of the first conference on First Workshop on Hot Topics in Understanding Botnets*, USENIX Association, Berkeley, CA, USA (2007), 4–4.

[SJVP+05] R. Sailer, T. Jaeger, E. Valdez, R. Perez, S. Berger, J. L. Griffin, L. van Doorn: Building a MAC-Based Security Architecture for the Xen Open-Source Hypervisor. *In: ACSAC '05: Proceedings of the 21st Annual Computer Security Applications Conference*, IEEE Computer Society (2005), 276–285.

[SZJv04] R. Sailer, X. Zhang, T. Jaeger, L. van Doorn: Design and Implementation of a TCG-based Integrity Measurement Architecture. In: *13th Usenix Security Symposium, San Diego, California* (2004), 223–238.

[tNA] trusted Network Access Control (tNAC). `http://www.tnac-project.org`.