Sicherheit in Informationssystemen

Fachhochschule Offenburg
Studiengang Medien und Informationswesen

Dipl.-Ing. (FH) Klaus J. Müller
kjm@internet-sicherheit.net

Inhalt

1 Grundlegendes / Denkanstöße

- Es gibt keine 100%ige Sicherheit
- Sicherheit läßt sich für die meisten nicht-trivialen Systeme nicht beweisen
- Software enthät Fehler! Mehr Code -> mehr Fehler
  -> kann zusätzliche Software helfen, das Sicherheitsniveau zu erhöhen?
- Das Wissen über Sicherheit muss ständig aktualisiert werden
- Das Sicherheitsrisiko ist proportional zum Produkt aus der Wahrscheinlichkeit des Eintretens eines Ereignisses und der Schwere der resultierenden Konsequenzen
- Der Aufwand muss proportional zum Risiko sein
- Sicherheit ist kein Zustand, sondern ein Prozess
- Sicherheit hat nicht nur technsche Aspekte: die Mitarbeiter sind ein Teil des Ganzen
- nicht auf „security by obscurity" vertrauen!
- These:

  Je geheimer ein Verschlüsselungs-Methode ist, umso sicherer ist das System

- Gegenthese:

  Je geheimer eine Verschlüsselungs-Methode, umso wahrscheinlicher, dass sie nur von wenigen durchdacht ist d.h. fehleranfällig ist

- besserer Ansatz:

  Bekannte und verstandene Algorithmen, die ausgereift sind

- Bei der Verschlüsselung geht man davon aus, dass die Leistung der Rechenwerke nicht sprunghaft ansteigt
- Sicherheit und Komfort/Benutzbarkeit stellen oft unvereinbare Maxime dar
  Designfrage: mehr Sicherheit oder mehr Komfort?

2 Voraussetzungen

2.1 Gefährdungen

Was ist gefährdet?

Wodurch?

2.2 Schützenswerte Eigenschaften

2.2.1 Lesen (passiv)

2.2.2 Verfälschung (aktiv)

2.2.3 Generieren (aktiv)

C	Confidentiality	Vertraulichkeit
I	Integrity	Integriät
A	Authenticity	Authentizität (Echtheit)
NR	Non – Repudiability	Nicht-Abstreitbarkeit

3 Security Policy

3.1 Authentifizierungsmechanismen

3.2 Physikalische Sicherheit

3.3 Passwort-Regeln

3.4 Firewall

3.5 verschlüsselte Kommunikation

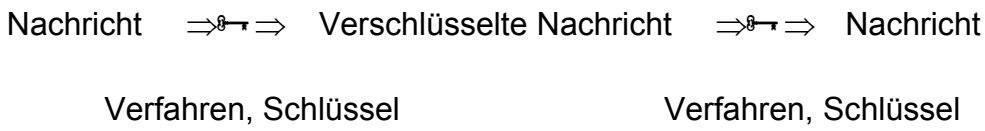
3.6 Dokumentation der Security Policy

3.7 Informierung der Mitarbeiter

3.8 Schulung der Administratoren und Benutzer

4 Kryptographie

Wissenschaft zur Erforschung und Realisierung von Verfahren zur Verschlüsselung bzw. Entschlüsselung von Daten, bei denen entweder das Verschlüsselungsverfahren oder (bei Anwendung einheitlicher Schlüsselverfahren) die verwendeten Schlüsselbegriffe geheim gehalten werden. Durch Ändern, Vertauschen oder Hinzufügen von Zeichen nach bestimmten Regeln wird ein Klartext in einen Schlüsseltext verwandelt und umgekehrt; anwendbar bei der Speicherung von Daten und der Datenübertragung . Wirksamstes Mittel des Datenschutzes, um Informationen, die in falsche Hände gelangt sind, wertlos zu machen. Neben der Vertraulichkeit durch die Verschlüsselung von Klartext kann mit den Methoden der Kryptographie auch die Authentizität einer Nachricht sowie die Integrität einer Datei sichergestellt werden, wobei unter letzterem, der Integrität einer Datei, die Gewissheit zu verstehen ist, eine Datei in unveränderter Form zu empfangen. Man unterscheidet bei den Verschlüsselungsverfahren ganz allgemein zwischen den asymmetrischen Verfahren (Public-Key) und dem symmetrischen. An bekannten Verschlüsselungsalgorithmen sind zu nennen: RSA, DES, 3DES, IDEA und AES.



4.1 symmetrische Verschlüsselung

Schlüssel sind auf beiden Seiten identisch.

Beispiel: XOR-Verknüpfung als Verschlüsselungsverfahren

PROBLEME:

- bei kurzen Schlüsseln kann aufgrund der Häufigkeitsverteilung der chiffrierten Daten auf den ursprünglichen Text geschlossen werden.
- Shared Secret - der Schlüssel ist auf beiden Seiten bekannt – entweder „shared" oder „secret"

LÖSUNGSANSATZ:

Man wählt den Schlüssel möglichst lang (im Idealfall so lang wie der zu verschlüsselnde Text).

4.2 asymmetrische Verschlüsselung (public key cryptography)

Voraussetzung für asymmetrische Verschlüsselung: Jeder Teilnehmer hat ein eigenes Schlüsselpaar:

geheimer Schlüssel: 

öffentlicher Schlüssel:

Mögliche Anwendungen:

- verschlüsseln ⇒Wahrung der Vertraulichkeit (C)
- signieren ⇒Wahrung der Integrität (I), der Authentizität (A) und der Nicht-Abstreitbarkeit (NR)

Pro Schlüsselpaar:

Ein Schlüssel für die Verschlüsselung:	öffentlicher Schlüssel des Empfängers
Ein Schlüssel für Entschlüsselung:	geheimer Schlüssel des Empfängers

Im Falle der digitalen Signatur:

zur Signatur:	geheimer Schlüssel des Absenders
zur Überprüfung der Signatur:	öffentlicher Schlüssel des Absenders

Beispiel: RSA-Algorithmus

1. Suche nach 2 Primzahlen	p,q
2. Bildung des Produkts	n= p*q
3. Bestimmung von $\phi(n)$	$\phi(n) = (p-1)*(q-1)$
4. Wählen von e als relative Primzahl zu $\phi(n)$	ggT (e, $\phi(n)$) = 1
5. Bestimmung von d so dass	$(d*e) \bmod \phi(n) = 1$
6. öffentlicher Schlüssel	e, n
7. geheimer Schlüssel	d, n

C	=	Chiffrat
e,n	=	öffentlicher Schlüssel
N	=	Nachricht

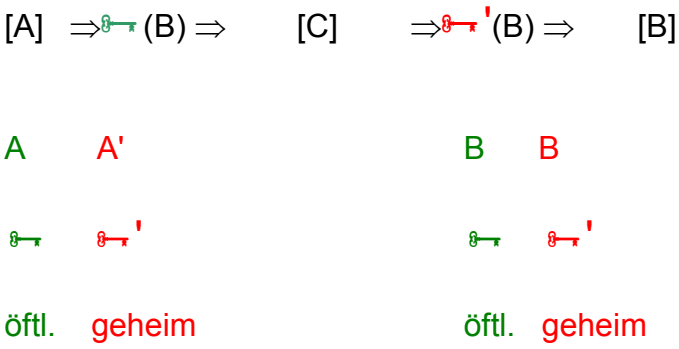
A) Verschlüsseln: $C = N^e \bmod n$
Entschlüsseln: $N = C^d \bmod n$

B) Signieren: $C = N^d \bmod n$
Überprüfen: $N = C^e \bmod n$

4.2.1 Verschlüsselung

Jeder der beiden Kommunikationspartner hat zwei Schlüssel – einen öffentlichen und einen geheimen. Der Partner, der sendet verwendet zur Verschlüsselung den öffentlichen Schlüssel des Gegenübers. Nur dieser hat dann die Möglichkeit die verschlüsselte Nachricht C wieder zu entschlüsseln, da nur sein geheimer Schlüssel passt.

Ziel: Nur Empfänger kann entschlüsseln: Confidentiality (Vertraulichkeit)



4.2.2 digitale Signatur

Bei der digitalen Signatur wird ein sogenannter Hash (H) zur Kontrolle verwendet. Aus der ursprünglichen Nachricht wird mittels eines Codierungsverfahrens der Hash gebildet. Dieser wird dann an die Nachricht gehängt. Der Empfänger ermittelt bei Erhalt den Hash aus der Nachricht und den Hash aus dem verschlüsselten Hash.

Stimmen beide Ergebnisse überein ist der Absender eindeutig identifiziert worden.

Ziel: Kontrolle, ob Nachricht verändert wurde (Integrität)

[A] ⇒ [N, H'] (Nachricht inkl. verschl. Hash) ⇒[B]

[N] ⇒ [H] ⇒🔑'(A) ⇒ [H'] [N] ⇒[H]

muss gleich sein

[H'] ⇒🔑(A) ⇒[H]

N ... Nachricht
H ... Hash, der aus Nachricht gebildet wird.
H' ... verschlüsselter Hash

4.2.3 Verschlüsselung (Real: Kombination den beiden vorherigen)

Ziel: Wahrung von Vertraulichkeit, Integrität, Authentizität und Nicht-Abstreitbarkeit (C, I, A und NR)

[A] [N] ⇒🔑(B) ⇒ [C] ⇒ [H(C)] ⇒🔑'(A) ⇒ [H'(C)]

[A] ⇒[C, [H'(C)]] Hashs müssen gleich sein

[B] [C, [H'(C)]] [H'(C)] ⇒🔑(A) ⇒H(C) C ⇒H(C)

[C] ⇒🔑'(B) ⇒[N]

Aufgrund von Performance wird bei der Kommunikation oft nicht jeder Datenaustausch über asymmetrische Verschlüsselung (sicher, aber zeitintensiv) realisiert, sondern nur einmal mit asymmetrischer Verschlüsselung ein Schlüssel übergeben, dann symmetrisch mit dem verschicktem Schlüssel verschlüsselt. In regelmäßigen Abständen wird dann immer wieder ein neuer asymmetrisch verschlüsselter Schlüssel verschickt, der dann für die symmetrische Verschlüsselung verwendet wird.

Alice [A] Bob [B]

🔑' 🔑 🔑' 🔑

shared secret für symmetrische Verschlüsselung — Verschlüsseln Signieren ⇒🔑(B) ⇒🔑'(A) [C, H] ⇒🔑(A) ⇒🔑'(B) ⇒ — Verifizieren Entschlüsseln — shared secret für symmetrische Verschlüsselung

4.2.4 Verteilung öffentlicher Schlüssel

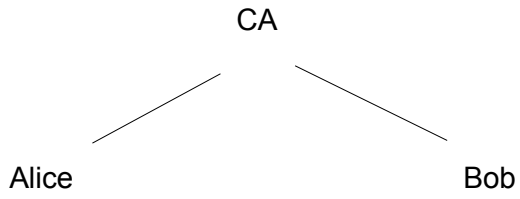
4.2.4.1auf dem physikalischen Weg

Die Schlüssel werden auf dem materiellen Wege ausgetauscht z.B. auf einem Datenträger oder auch schriftlich/persönlich

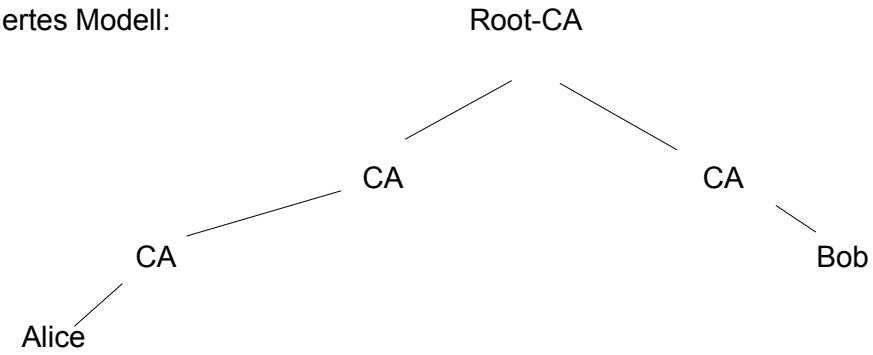
4.2.4.2PKI Public Key Infrastructure

Eine Zertifizierungsinstanz (Certification Authority, CA) bestätigt, dass hinter einer bestimmten Signatur (einem öffentlichen Schlüssel) tatsächlich eine bestimmte Person steckt.

Grobes Modell:



Verfeinertes Modell:

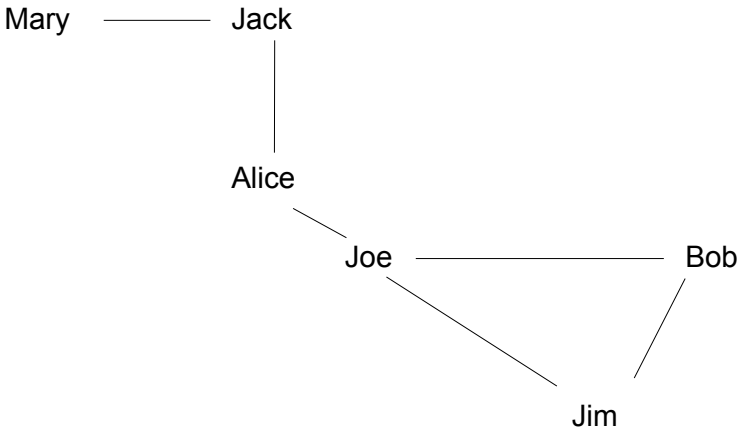


Vorteil: Hierarchische Struktur ist kontrollierbarer als Web of Trust

Nachteil: hohe Startinvestition, eine solche Instanz aufzubauen

4.2.4.3Web of Trust (Vertrauensgewebe)

Hierbei bestätigt eine Person einer anderen (ihr bekannten Person) die Echtheit der Signatur. Mary kennt Bob, Bob kennt Jim usw. Es gibt jedoch keine Hauptinstanz!



Vorteil: keine hohe Startinvestitionen, erlaubt komplexe Vertrauensbeziehungen

Nachteil: schwer steuerbar: z.B. Schlüssel ungültig werden lassen.

4.2.5 Anwendungen von Verschlüsselung

4.2.5.1auf Anwendungsebene (Bsp 1): Mail

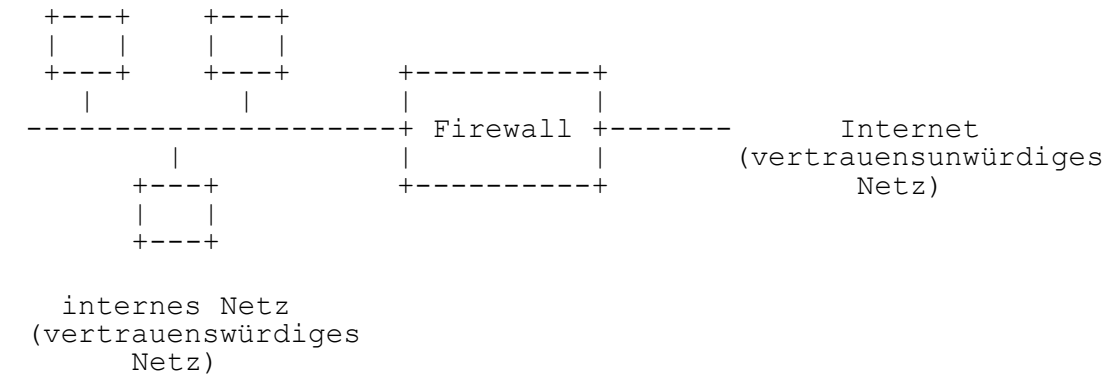
4.2.5.2auf Anwendungsebene (Bsp 2): Web

4.2.5.3auf Netzebene

5 Technische Lösungen

5.1 Firewalls

Allen Firewalls gemeinsam ist, dass sie an der Schnittstelle zwischen Netzen sitzen. Sie dienen dazu, ein als unsicher eingestuftes Netz von einem sicheren zu trennen. Sämtliche Kommunikation zwischen den beiden Netzen läuft über die Firewall ab. Die Firewall trifft nun anhand der ihr zur Verfügung stehenden Informationen über die jeweilige Verbindung und eines Regelsatzes die Entscheidung, ob eine Verbindung zugelassen wird oder nicht.



Für alle Firewalls gilt, dass sie nach einer von zwei verschiedenen Grundstrategien verfahren:

Alles, was nicht explizit verboten ist, ist erlaubt

oder:

Alles, was nicht explizit erlaubt ist, ist verboten.

5.1.1 Einschub: Grundlagen TCP / IP

4 Schichten:

4 Anwendungsschicht	(z.B. FTP, http, POP3, SMTP)	
3 TCP / IP	(IP Adressen, Ports)	
2 IP	(IP – Adressen)	
1 Netzzugangsprotokoll	Ethernet	Modem, PPP
	[MAC Adresse]	[Telefonnummer]

5.1.1.1Verbindungsaufbau mit TCP

1. Paket mit „ SYN“ - Flag vom Client zum Server
2. Paket mit „ SYN“ - Flag und „ACK“-Flag vom Server zum Client
3. Paket mit „ ACK“ – Flag vom Client zum Server
4. folgende Pakete mit „ACK“ – Flag in beide Richtungen

5.1.1.2Kommunikation auf Anwendungsebene

z.B. Anfrage des Client -> GET/

z.B. Antwort des Servers -> HTML – Dokument

5.1.2 Typen von Firewalls

Die verschiedenen Firewall-Typen unterscheiden sich in der Information, die ihnen zur Verfügung stehen und in der Schicht, auf der sie arbeiten.

5.1.2.1Paketfilter

Ein Paket-Filter arbeitet auf Schicht 3 des TCP-Modells. Ihm sind daher die IP-Adressen und Portnummern von Sender und Empfänger sowie verschiedene Flags bekannt. Anhand dieser Parameter kann er seine Entscheidung treffen.

Beispiel-Regelsatz:

Voreinstellung: defensive Strategie (alles, was nicht explizit erlaubt ist, ist verboten).

Quell-IP	Quell-Port	Ziel-IP	Ziel-Port	Flags
Web:				
192.168.0.0/24	1024-65535	0.0.0.0/0	80	(beliebig)
0.0.0.0/0	80	192.168.0.0/24	1024-65535	ACK
Mail (POP3):				
192.168.0.0/24	1024-65535	0.0.0.0/0	110	(beliebig)
0.0.0.0/0	110	192.168.0.0/24	1024-65535	ACK

Vor-/Nachteile von Paket-Filter:

+ anwendungsunabhängig
+ performant
+ skalierbar
- niederste Sicherheitsstufe

- kein Wissen von Applikationen der Schicht oberhalb von TCP und Vorgänge in deren Protokollen

5.1.2.2Stateful Inspection

Dieser erweiterte Paketfilter hält eine Liste der aktuellen Verbindungen. Der jeweilige Status einer Verbindung spielt in der Entscheidung eine Rolle. Definition der Filterregeln besteht nur aus den Anfragen - entsprechende Antworten werden automatisch zugelassen

- alle Pakete, die als Antwort zu einer Anfrage kommen, werden automatisch durchgelassen

Beispiel-Regelsatz:

Voreinstellung: defensive Strategie (alles, was nicht explizit erlaubt ist, ist verboten).

Quell-IP	Quell-Port	Ziel-IP	Ziel-Port	Flags
Web:				
192.168.0.0/24	1024-65535	0.0.0.0/0	80	SYN
Mail (POP3):				
192.168.0.0/24	1024-65535	0.0.0.0/0	110	SYN

Vor-/Nachteile von Stateful Inspection:

+ anwendungsunabhängig
+ performant
+ skalierbar
+ erlaubt einfachere (und damit besser wartbare) Regelsätze
+ bestimmte Angriffsversuche werden abgewehrt
- kein Wissen von Applikationen der Schicht über TCP und Vorgänge in deren Protokollen

5.1.2.3Application Proxies

Diese arbeiten auf Schicht 4 des TCP-Modells und „sprechen“ daher das Anwendungsprotokoll (z.B. HTTP, FTP, SMTP...). Eine Filterung kann anhand der Kommandos des Anwendungs-Protokolls erfolgen.

Der Proxy akzeptiert eine Verbindung (von intern) auf der einen Seite und baut auf der anderen eine separate (nach extern) auf. Dadurch fließt der Verkehr nie direkt zwischen Client und Server. Er kann Entscheidungen auf Applikationsebene treffen, die größere Flexibilität ermöglichen, welche Art von Verkehr zugelassen wird.

Ausser den protokoll-spezifischen Proxies (HTTP, FTP, SMTP) gibt es noch generische Proxies. Diese werden eingesetzt, um Proxy-Dienste für Protokolle anzubieten, für die kein spezifischer Proxy existiert. SOCKS ist eine solche Implementierung. Hierbei wird auf dem Client eine Komponente in den Netzwerk-Treiber integriert, welche – transparent für die Anwendung – die Verbindung nicht direkt zum Server, sondern über den Proxy zum Server aufbaut.

Vor-/Nachteile von Proxies

+ Verstehen der Vorgänge in der Anwendungsschicht
+ zusätzliche Sicherheit
- weniger performant
- nicht transparent

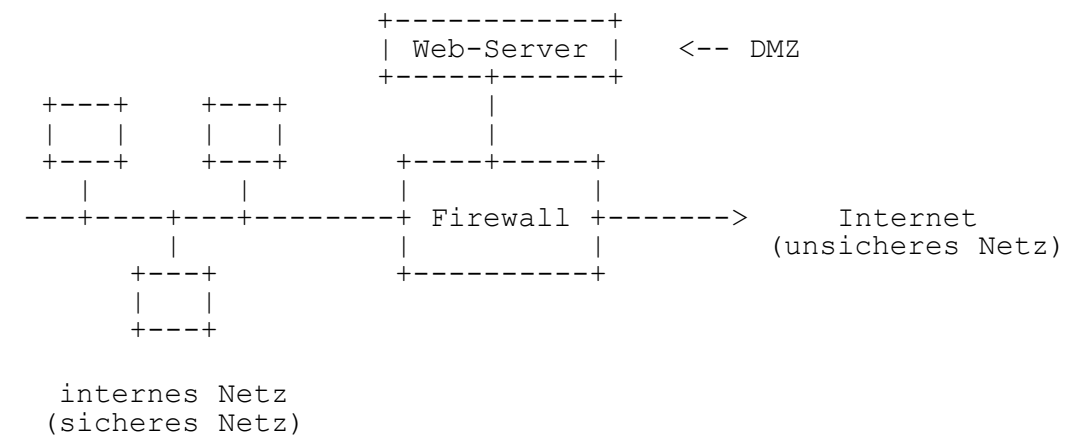
5.1.2.4Content Filter als Erweiterung Protokollspezifischer Application Proxies

Was wird gefiltert ?

- Kommandos (protocol spezifisch, z.B. GET zulassen PUT nicht zulassen)
- Inhalt (Virenscanner)
- Zieladressen (z.B. realtime block lists – liste mit Seiten, die nicht geliefert werden sollen – werden in Echtzeit (z.B. alle 10 Min) aktualisiert)

5.1.3 Demilitarisierte Zone

Rechner, die auch aus dem Internet erreichbar sein sollen, werden oft in einem eigens dafür eingerichteten Netz aufgestellt: in der Demilitarisierten Zone (DMZ). Zugriffe aus dem Internet auf das interne Netz können nun weiterhin vollständig unterbunden werden. Zugriffe aus dem Internet auf die Rechner in der DMZ (z.B. Web-, FTP-, Mail-Server...) können hingegen – sofern sich nicht weitere Reglen dies verbieten – durchgelassen werden.



Für den Fall, dass es einem Hacker gelingt, eine Attacke auf die Rechner in der DMZ zu starten, wird das interne Netz noch immer geschützt.

Zur Realisierung einer DMZ benötigt man 2 Firewalls mit je zwei Netzwerkschnittstellen oder eine Firewall mit 3 Netzwerkschnittstellen.

5.2 Intrusion Detection (IDS)

Eine Intrusion Detection (Einbruchserkennung) ist gleichzusetzen einer Alarmanlage in der Systemkonfiguration. Man kann zwischen 2 Systemen unterscheiden:

- hostbasiertes System: wird auf dem lokalen Rechner installiert
- netzbasiertes System: wird in dem Netzwerk des zu überwachenden Rechners betrieben

2 Ansätze:

5.2.1 Signaturerkennung

Es ist eine Signatur hinterlegt, die Zustände definiert, die bedenklich sind.

Beispiele:

Login zu ungewöhnlichen Zeiten in ungewöhnlicher Anzahl (mehrere FTP-Logins gleichzeitig für einen User ...; Netzbasierte Anomalie Erkennung)

Vorbereitung eines Angriffs durch Scannen der Ports; Hostbasierte Anomalie Erkennung überwacht Ports

➡ Folglich: Festlegung von außergewönlichen Ereignissen (= Signaturen), auf die in bestimmter Weise reagiert werden soll

5.2.2 Anomalieerkennung

Zuerst muss das System „lernen“ was die Normalkonfiguration, bzw. das Normalverhalten der Anwendungen ist. Dann wird bei Abweichungen davon (= Anomalie) ein Alarm ausgegeben.

PROBLEM: Eine exakte Einstellung zwischen möglichst großer Fehlererkennung und andauernder Fehlalarmierung ist sehr schwer und verlangt vom Administrator hohes Wissen und Aufmerksamkeit.

Die beiden Systeme (host- und netzbasiert) lassen sich mit den beiden Ansätzen (Signatur- und Anomalieerkennung) kombinieren

wichtige Begriffe im Kontext von IDS:

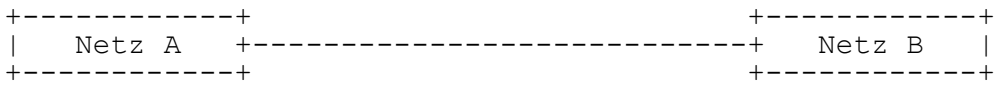
- false positive

- false negative

5.3 Virtual private Network (VPN)

Zur Verbindung mehrerer geografisch verteilter Firmenstandorte gibt es mehrere Ansätze.

5.3.1 klassische Lösung der Verbindung mehrerer Standorte über Wähl- oder Standleitung



Standort A (öffentliches Telefon-Netz) Standort B

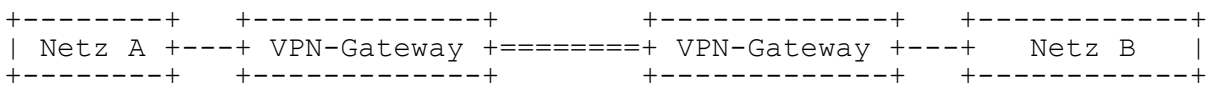
PROBLEME:

- im Normalfall nicht verschlüsselt
- keine Authentifizierung des „Anrufers“
- Leitung kann abgehört werden
- hohe Betriebskosten (Leitung)

5.3.2 VPN

Bei einem Virtual Private Network (VPN) werden die Pakete über eine verschlüsselte Verbindung über das Internet übermittelt. Durch die Verwendung einer normalen Internet-Verbindung werden die Kosten minimiert (Leitung nur noch vom Standort zum Provider). Gleichzeitig wird durch die Verschlüsselung C, I und A gewährleistet.

5.3.2.1Netz-zu-Netz Szenario

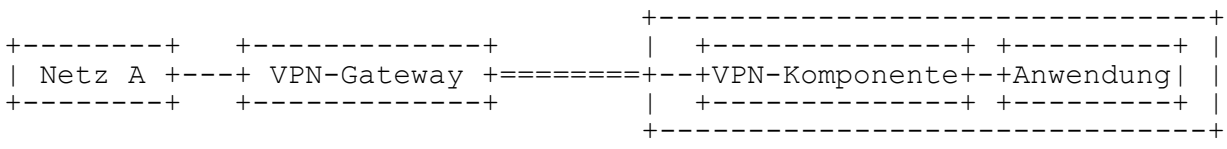


Standort A (Internet) Standort B

Þ verschlüsselte Verbindung

➫ verschlüsselte Verbindung durch Tunneln des Netzwerk-Verkehrs

5.3.2.2Host-zu-Netz Szenario



Standort A (Internet) mobiles Endgerät
(Road Warrior)

5.4 Redundanz

5.4.1 Auf Komponenten-Ebene

5.4.2 Auf Rechner-Ebene

5.4.3 Auf Dienste-Ebene

6 Verfahren

6.1 Authentifizierungsmechanismen

6.1.1 Wissen

Verfahren, die darauf basieren, dass ein berechtigter Benutzer etwas bestimmtes Wissen muss: Passwort, PIN, VISA-Karteninformationen

6.1.2 Besitz

In diesem Fall ist der Besitz eines bestimmten Gegenstandes erforderlich: Kontokarte, Smartcard, USB-Token, VISA-Karte

6.1.3 Unablegbare Eigenschaft: biometrische Verfahren

Hier wird der Benutzer auf eine unablegbare Eigenschaft überprüft: Fingerabdruck, Iris-Muster, Stimme, DNA

6.2 Härtung von Betriebssystemen

1. nur die minimale Installation (SW Pakete), die für den Betrieb des Computers unbedingt notwendig ist.

2. Installation aller verfügbaren Patches für alle installierten Pakete (sollte als Teil der OS Installation betrachtet werden)

3. Installation der neuesten verfügbaren stabilen Versionen der verwendeten Produkte.

4. Entfernen aller Privilegien und Zugriffsrechte. Anschließende Gewährung von Zugriffen wo es unbedingt erforderlich ist.

5. Aktivierung der höchstmöglichen Protokollierung um möglichst detaillierte Informationen zu erhalten, die im Falle eines Einbruchs für eine exakte Analyse benötigt werden.

ZIEL: Minimieren der Angriffsfläche

6.3 Open Source

6.3.1 Code Review

Es besteht die Möglichkeit Fehler und absichtlich eingebaute Hintertüren aufzuspüren und zu eliminieren. Die Wahrscheinlichkeit ist sehr hoch, dass existierende Sicherheitslücken entdeckt werden.

- Zuverlässigkeit
- Stabilität
- „bad code“ hat keine Chance

6.4 Security Audit

Analyse der Sicherheitssituation

Anwendbar auf:

- Software
- Netzwerkinstallationen

6.4.1 Varianten der Umsetzung

- Black Box

Sicherheitsanalyse wird durchgeführt durch ein externes Unternehmen. Dieses Unternehmen hat keine Informationen über die Infrastruktur. Situation eines potenziellen Angreifers.

- White Box

Sicherheitsanalyse wird im Unternehmen selbst durchgeführt. Es wird mit Hilfe aller Informationen getestet wie sicher das System ist. Die Analyse sollte nicht vom Administrator selbst vorgenommen werden.

6.5 Penetrationstests

7 Malicious Software oder Malware

7.1 Hintertüren (Backdoors)

aus Entwicklersicht (“gute Beweggründe“) - um im nachhinein Änderungen an der Software vorzunehmen oder um Wartungsarbeiten an der Software durchzuführen. Realisiert durch „Generalschlüssel“.

Im einzelnen werden folgende Gründe genannt:

7.1.1 Umgehung von Authentifizierung

7.1.2 Erlangung von besonderen Privilegien

7.1.3 Fehlersuche / Problembehandlung wenn die Authentifizierung nicht funktioniert

7.2 trojanische Pferde

Enthalten ausser einer vordergründig nützlichen Funktionen für den Anwender Schadensfunktionen oder Spionagesoftware.

Beispiel: Tool bei T-Online, das Passwörter verschickt hat.

7.3 Viren

Programmteil welcher andere Programme infizieren kann, indem es diese verändert.

wesentliche Eigenschaften:

- kein eigenständiges Programm, sondern Code, der sich an bestehenden Code „anheftet“
- Replikation (erstellt Kopien von sich selbst)

Die verbreitetsten Arten von Viren sind:

7.3.1 „Binär-Viren“

7.3.2 Boot – Sektor – Viren

7.3.3 Makro Viren

7.3.4 Einschub:

Programme wie Outlook Express bieten viele Features, die es dem User einfach machen sollen, aber letztendlich auch ein hohes Sicherheitsrisiko darstellen (Frage: Pro Feature oder pro Sicherheit?) !

Beispiele:

1.verwendete Dateierweiterungen werden nicht angezeigt, so heißt eine Datei datei.txt.exe im Outlook nur datei.txt !

2.E-Mail-Anhänge werden schon beim Betrachten der E-Mail „gestartet“!

7.3.5 Hoax (Scherz)

7.4 Würmer

Im Gegnsatz zu einem Virus ist ein Wum ein eigenständiges Programm. Genau wie ein Virus hat auch ein Wurm Funktionen zur Selbstreplikation.

7.5 Root-Kits

7.6 Abhilfe

7.6.1 Virenscanner

7.6.2 sichere Konfiguration der Software

7.6.3 Content (Inhalt) Filter

7.6.4 Schulung der Anwender

8 Analyse Tools

8.1 Sniffer

8.2 Port Scanner

8.3 Honeypots

9 Steganographie

Ziel: Kommunikation oder Daten verstecken. Diese Daten können zusätzlich wiederum verschlüsselt sein, müssen es aber nicht.

10 Rechtliche Aspekte

11 Literatur

Sicherheit im Internet
William Stallings
ISBN: 3-8273-1697-9

Hacking Exposed
Joel Scanbury/Stuart Mc Clare/George Kurtz
ISBN: 0-07-212748-1

Applied Cryptography
Bruce Schneier
ISBN: 0-471-11709-9

The CERT Guide to System and Network Security Practices
Julia H. Allen
ISBN: 0-201-73723-X

Linux Hacker's Guide
Anonymous
ISBN: 3-8272-5622-4

Secrets & Lies
Bruce Schneier
ISBN: 0-471-25311-1

12 Links

http://www.cert.org/	
http://www.securityfocus.com/	
http://www.sicherheit-im-internet.de/	Seite des BSI
http://www.ntbugtraq.com/	speziell für Win NT
http://www.heise.de/ct/pgpCA/	c't-Krypto-Kampagne
http://www.heise.de/ct/antivirus/	c't-Antivirus-Informationen
http://www.pro-privacy.de/weboftrust.htm	Erklärungen zum Web-of-Trust

13 Software

Passwort-Verwaltung:

- Password Keeper von Wolff-Software
  http://www.wolff-software.de/
- FPM (Figaro's Password Manager) für Linux
  http://fpm.sourceforge.net/
- Steganos Password Manager
  http://www.steganos.com/de/pwdm/index.htm
- Password Safe von Counterpane
  http://www.counterpane.com/passsafe.html

Verschlüsselungssoftware:

- PGP (Pretty Good Privacy)
  http://www.pgpi.org
- GnuPG oder GPG – Alternative zu PGP (wird unterstützt vom BSI – Bundesministerium für Sicherheit in Informationssystemen)
  http://www.gnupg.org

Sniffer:

- Ethereal
  http://www.ethereal.com/

Portscanner:

- Nmap
  http://www.insecure.org/nmap/

Steganographie-Software:

- Steganos Security Suite (Test-Version)
  http://www.steganos.com/de/sss/index.htm
- weitere:
  http://www.heise.de/ct/01/09/links/170.shtml