Firewall

My Company - My Castly

Kontinuierlicher Prozess

- Im Idealfall sollte (!) IT-Sicherheit ein kontinuierlicher Prozess aus folgenden Stufen sein
- Protection Phase
- Detection Phase
- Response Phase

Protection Phase

- Aus den Sicherheitanforderungen - abgeleitet aus der Security Policy und den Sicherheitskonzepten - werden Massnahmen definiert.
- Diese Massnahmen werden sukzessive umgesetzt um zu einem Zeitpunkt X einen definierten - sicheren - Zustand der Systeme zu erreichen

Detection Phase

- Durch den Einsatz von Host - oder Netzwerk basierten Intrusion detection Systemen werden die Angriffe auf das Systeme überwacht und ausgewertet (!)
- Insbesondere Auswertung und Überwachung ermöglichen eine Ausgangsbasis für die 3. Phase

Response Phase

- Durch Erkenntnisse die aus der Überwachung gewonnen werden, kann das Sicherheitssystem permanent aktualisiert werden
- So wird auf neue Angriffe, Schwachstellen in Betriebssystemen oder neue Angriffstechniken reagiert
- Diese 3 Phasen werden permanent durchlaufen und ggf. aktualisiert

Maßnahmen

- Intrusion Detection
  - Host - oder Netzwerkbasiert
- Firewallsysteme
  - ggf. auch durch DMZ getrennt
- Maßnahmen auf Betriebssystemebene
  - insbesondere auf Servern die für externe Aufgaben zur Verfügung stehen (z.B. WWW Server)

Intrusion Detection Systeme

- Werden mit Netzwerk - oder Host-Sensoren ausgeliefert
- Diese Sensoren haben eine Policy
- Je nach Policy werden definierte Ereignisse an eine Datenbank geschrieben, der Port wird gesperrt, die IP wird gesperrt, der Admin erhält eine Mail und/oder der Vorstand eine SMS usw.....
- Produkte: Real-Secure (ISS), Dragon, Snort

Bsp ISS

- Sensoren für Solaris, W2K und Linux
- Wird mit vordefinierten Policys ausgeliefert
  - Diese sind natürlich konfigurierbar !
- Alle Sensoren liefern ihre Informationen an einen Event Collector (Wehe, wenn der ausfällt)
- Dieser schreibt die Informationen in eine Datenbank (2 GB wird mitgeliefert)

Betriebssystemebene

- Minimalprinzip
- Nutzung minimaler Rechte und entsprechender Kernel
- Deaktivierung nicht benötigter Dienste
- Aktualisierung durch entsprechendes Patchen

Möglichkeiten

- Scannen eines Hosts mit z.B. nmap
  - welche Ports sind offen ?
  - Nötig oder nicht ?
- Software auf einem Server
  - Sind Entwicklungsumgebungen auf einem Server notwendig ?
  - Gcc, Skriptsprachen ?

Firewalltypen

- Applicationlevel Gateway
  - Proxies [Vertreter] -> Programme die auf Applicationsebene auf einen bestimmten Dienst spezialisiert sind. Jede Verbindung aus dem zu schützenden Netz wird an den Proxy gerichtet, der gibt die Anfrage an das unsichere Netz weiter.

Vorteil - Nachteil

+ Kann auf Applicationsebene absichern
+ Nutzdaten werden erfasst
+ funktioniert auch als Cache für Seiten
+ Kann als Reverse - Proxy Rechenleistung übernehmen

- benötigt mehr Rechenleistung als Paketfilter bei entsprechenden Bandbreiten
- meist schwieriger zu konfigurieren

Firewalltechniken

- Paketfilter
  - Netzwerkebene
  - IP Datagramme werden in das Netz hinein und aus dem Netz heraus überprüft
  - Filterregeln treffen die Entscheidung, ob ein Paket durchgelassen wird oder nicht (drop, reject)

Was wird überprüft ?

- Reiner Paketfilter OSI Level 3 und 4 (Network und Transport)
- Grundsätzlich wird nur der Header betrachtet
- Vorteile:
  - "Relativ" einfache Konfiguration an einem festgelegten Punkt
  - Schnelle Entscheidungen wenig Performance Verluste

Nachteile

- Nur in begrenztem Umfang können
  - Quell - und Zielport
  - u. Protokolle überprüft werden
- ICMP Typen
- Aber:
  - Keine Nutzdaten wie Viren, Kontraproduktive Programme oder ungewollte Inhalte

Paketfilter

- Ein Paketfilter kontrolliert und analysiert den ein- und ausgehenden Datenstrom auf der Netzzugangs-, der Netzwerk- und der Transportebene. Die Analyse wird derart durchgeführt, daß die Datenpakete, die durch das physikalische Kabel übertragen werden, aufgenommen und analysiert werden

Arbeitsweise

- Von welcher Seite - des Netzwerks - kommen die Pakete
- Protokolltyp Ziel - und Quelladresse
- Portnummer
- ggf. Zeitraum
- Verstoßen die hier gefundenen Informationen nicht gegen die aufgestellten Regeln werden die Pakete durchgelassen, sonst .. Je nach Konfiguration

Bauarten

- Router
  - Router übernimmt Firewallfunktionalitäten
  - Vorteil: Nicht noch ein Bauteil ...
  - Nachteil: Einschränkungen bei Protokollierung, kompliziertere Programmierung der Filterregeln

Bauarten

- Seperater Paketfilter
- Auf Softwarebasis - z.B. Netfilter/Iptables - die Software übernimmt die Funktionalität
- Vorteil: Leichtere Programmierung der Filterregeln, bessere Programmierung der Meldungen an den Admin bei Alarm
- Gelegentlich teurer

NAT (PAT)

- Network Adress Translation
- Im internen Netz wird ein privater Adressraum genutzt
- Der Paketfilter tritt als eine - öffentliche - IP auf, und kann so das interne Netz vor den neugierigen Blicken aus dem öffentlichen Netz verbergen

Filterregeln

- Zwei grundsätzliche Ansätze:

"Es wird alles erlaubt, was verboten ist wird deaktiviert"

"Es wird alles verboten, nur was explizit erlaubt ist, wird aktiviert"

Filterregeln an einem Beispiel

```
#!/bin/sh

# ISDN
INTERFACE=ippp0
# Modem
#INTERFACE=ppp0
# Ethernet (eth0=LAN, eth1=DMZ, eth2=Internet)
#INTERFACE=eth2

insmod ip_tables
insmod ip_conntrack
insmod ip_conntrack_ftp
insmod ipt_state
insmod iptable_nat
insmod ipt_MASQUERADE
```


Filterregeln an einem Beispiel

```
iptables -F

iptables -N block
iptables -A block -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -A block -m state --state NEW -i ! $INTERFACE -j ACCEPT
iptables -A block -j DROP

iptables -A INPUT -j block
iptables -A FORWARD -j block

iptables -A POSTROUTING -t nat -o $INTERFACE -j MASQUERADE
echo 1 > /proc/sys/net/ipv4/ip_forward
#Quelle: Linux Magazin
```


Erläuterung

```
#!/bin/sh

# ISDN
INTERFACE=ippp0
#dev wird bestimmt, entweder ippp0 - ISDN oder entsprechende Moden
    Schnittstelle
#Danach werden mit insmod Module geladen, mit denen dann ein neuer
    Kernel kompiliert wird.

iptables -F
#bestehende Filterregeln werden gelöscht, wichtig, wenn Skript mehr als
    einmal aufgerufen wird..
```

Erläuterung

iptables -N julius

#es wird mit -N julius eine neue Chain angelegt.

Vorhandene Chains sind

- INPUT Pakete die an die Firewall selbst gerichtet sind
- FORWARD Pakete die an andere geschickt werden und von der Firewall nur weitergeleitet werden sollen
- OUTPUT ...

iptables -A julius -m state --state ESTABLISHED,RELATED -j ACCEPT

#Diese Regel wird mit der Options -A “Abandon” an die CHAIN “julius” angehängt

-m state

#Status des Pakets (Verbindungsorientierte Protokolle)

Erläuterung

--state ESTABLISHED,RELATED

ESTABLISHED Paket gehört zu einer bestehenden Verbindung

RELATED Paket gehört nicht zur selben Verbindung hat aber Beziehung dazu. Bsp. FTP

#Andere sind

#NEW Neue Verbindung

#INVALID Unindentifizierbare Pakete - werden nie weitergeleitet

-j ACCEPT

#Was mit dem Paket anstellen “-j” ?

#ACCEPT ..

#DROP - Verwerfen

#LOG - Loggen ..

Erläuterung

-A julius -m state --state NEW -i ! $INTERFACE -j ACCEPT

-i In interface

-o Out Interface

! $INTERFACE ist hier die entsprechende Variable

iptables -A julius -j DROP

Alle Pakete die nicht in diese Regeln passen, werden verworfen (“DROP”)

iptables -A INPUT -j julius

Alle Pakete die an die Firewall gesendet werden, sollen an die Chain “julius” übergeben werden- wie in einem Unterprogramm

Regeln für ganze Ketten

- Eine neue Kette erstellen (-N).
- Eine leere Kette loeschen (-X).
- Die Policy fuer eine eingebaute Kette aendern (-P).
- Die Regeln einer Kette auflisten (-L).
- Die Regeln aus einer Kette ausspuelen (flush) (-F).
- Paket- und Bytezaehler aller Regeln einer Kette auf Null stellen (-Z).

Regeln einer Kette manipulieren

- Eine neue Regel an eine Kette anhaengen (-A).
- Eine neue Regel an eine bestimmte Position in der Kette einfuegen (-I).
- Eine Regel an bestimmter Position in der Kette ersetzen (-R).
- Eine Regel an einer bestimmten Position in der Kette loeschen (-D).

Einige Flags

- -p Protokoll
- -s Source
- -d Destination
- ! Nicht
- .....
- Quellen:
  - http://www.netfilter.org/documentation/HOWTO/de/packet-filtering-HOWTO-7.html