

sigN
(sniffen in geschwichten Netzwerken)

DaNiel Ettle
FH-Regensburg
Informatikstudent

23.05.2002

Inhaltsverzeichnis

1	Einstimmung	3
2	Grundlagen und Voraussetzungen	4
2.1	Verantwortung	4
2.2	Hinweise	4
2.3	Grundlagen	4
3	Einfuehrung	5
3.1	allgemein	5
3.2	Schichtwerk	5
3.3	Hardwarezeich	5
4	Netztopologien	6
4.1	Shared Medium Netzwerke	6
4.1.1	Einfuehrung	6
4.1.2	Beispiel	6
4.2	geswitchte Netzwerke	6
4.2.1	Einfuehrung	6
4.2.2	Beispiel	7
5	Hijacking	7
5.1	ARP Spoofing/Relaying - Layer 2	7
5.1.1	Was'n ARP?	7
5.1.2	Kids-club	8
5.1.3	blonde Hardware	8
5.1.4	Gundel Gaukelei	8
5.1.5	Spoofen	8
5.1.6	Eisszeiten und Einfriermoeglichkeiten	9
5.2	TCP - Layer 4	9
5.2.1	Grundlegendes	10
5.2.2	Verbdingsaufbau...	10
5.2.3	... und Abbau	11
5.2.4	RFC 793, RST und andere Phaenomene	11
5.2.5	Desynchronized state	12
5.2.6	Angriff	13
5.2.7	ACK-Storm	14
5.2.8	Kevin vs. Tsutomu	14
5.2.9	Verteidigung	16
5.3	UDP - Layer 4	17
5.3.1	Flutwellen,	17
5.3.2	Stuerme	18
5.3.3	... und andere Katastrophen	18
5.3.4	Deiche, Daemme und brennende Waende	18
5.4	ICMP - Layer 3	19
5.4.1	RFC 792 und andere Zustaende	19
5.4.2	Router Advertisement	19
5.4.3	Redirection - Umleitung	20
5.4.4	Source Quench	20
5.4.5	Durstloescher und andere Schutzmassnahmen	20
6	Realitaetsabgleich	21
6.1	Szenario 1	21

7 Tools	21
7.1 do it yourself	21
7.2 ready, steady, hijack	22
8 Schutzmassnahmen	22
8.1 Einfuehrung	22
8.2 Hardware	22
8.3 Software	22
8.4 IP Filter	23
8.4.1 Inputfilterung	23
8.4.2 Empfaenger IP Filterung	23
8.4.3 AbsenderPort Filterung	24
8.4.4 EmpfaengerPort Filterung	24
8.4.5 TCP Flags	24
8.5 Umgang mit ICMP	24
8.6 misstrauen	25
8.7 wenss doch passiert?	26
9 Quellen	26
10 Danksagung	27

1 Einstimmung

Es passiert immer genau dann, wenn man meint, dass man nichts zu befürchten hat. Jahrelang dachte man, dass geschwichtete Netzwerke sicher vor sniffing-attacks sind. Viele wagten sich damit in Sicherheit.

Das Arsenal der Werkzeuge, die es für beide Seiten gibt, sowohl für das Eindringen, wie auch für das Schützen von Netzwerken, ist erschreckend gross. Sowohl die Anzahl, als auch die Qualität der Tools ist mittlerweile unüberschaubar geworden. Einige dieser Werkzeuge haben sich auf gewisse Bereiche spezialisiert und sind darin unschlagbar, andere wiederum sind Universal- oder Multifunktionswerkzeuge, bei denen man Plugins hinzufügen oder die man selber programmieren kann.

Network sniffers sind eigentlich aus völlig harmlosen Gründen entstanden, sie sollten den Netzwerktraffic analysieren und Schwachstellen oder Fehlkonfigurationen finden, um diese zu beheben. Ein sniffer macht eigentlich nichts anderes als Netzwerkframes abzufangen und diese darzustellen. Aus diesem harmlosen Tool für Administratoren, die damit ihr Netzwerk auf Probleme hin überprüfen wurde ein Programm, das Cracker verwenden, um Passwörter oder Daten aus der Leitung zu saugen.

Wenn man nun die Zeitspanne betrachtet, die zwischen dem ersten Auftreten von Artikeln und Tools und dem verbreitendem Bekanntwerden dieser Sicherheitslücken ist, verwundert es einen fast, dass nicht mehr passiert. Diese verzögerte Informationsverbreitung erhöht das Sicherheitsrisiko in grossem Masse. Nichts ist schlimmer, als zu glauben, dass etwas sicher ist, das gar nicht sicher ist.

IP-Spoofing und ARP-Relaying Attacken verhalfen unter anderem einem der weltberühmtesten Cracker namens Kevin 'Condor' Mitnick zu einem temporären Siegeszug gegen die grössten Provider, Hosts und Telefonanbieter Amerikas. Zur damaligen Zeit konnte sich diese fiktive Attacke erst in den Köpfen einiger Computer-Experten etablieren, und wurde noch nie in freier Wildbahn genutzt. Daher konnten sich die betroffenen, zuständigen Administratoren lange nicht erklären, wie Herr Mitnick sich unerlaubten Zugang zu vielen internen, eigentlich gut geschützten Systemen, hatte verschaffen können. Wer mehr über den Fall Kevin Mitnick wissen möchte, dem sei das Buch „Data Zone - Die Hackerjagd im Internet“ ans Herz gelegt. Der Film „Takedown“ handelt zwar ebenfalls von Kevin M., erzählt die Geschichte aber nur unzureichend.

Für viele ist das Herausfinden von Daten nur ein spassiger Zeitvertreib, wenn man das email-passwort seines WG-Kollegen rausbekommen will. Klar, das ist alles halb so wild und macht auch Spass, wenn man dann allerdings selber mal mit seinen geheimen persönlichen Daten konfrontiert wird, dann würde man sich doch sehr wünschen, dass alles etwas sicherer wäre. Selbst wenn man selber immer auf Sicherheit geachtet hat und alles unternommen hat, kann eine unachtsame Firma oder ein bekannter User, mit dem man des öfteren Kontakt hat, ein Sicherheitsproblem haben. Es hat sicherlich schon jeder mal irgendein Passwort plain-text übertragen, sei es per mail, telnet, irc oder sonstwie. Das ist schneller passiert als man denkt.

Wie die Geschichte zeigt, wird Sicherheit immer klein geschrieben, bis es irgendwo hackt(ed).

2 Grundlagen und Voraussetzungen

2.1 Verantwortung

Dieser Artikel soll zeigen, wie man in geschwichten Netzwerken mitsniffen kann und wie man sich dagegen schuetzen kann.

Jeder, der die Moeglichkeit besitzt, Daten zu sichten, die nicht fuer ihn bestimmt sind, sollte wissen was er tut und verantwortungsbewusst handeln.

Das Veraendern oder Loeschen von Daten sowie das weitergeben von Daten an Dritte ist zwar in einigen Laendern – wie Argentinien – noch nicht strafbar, hier in Deutschland allerdings muss man mit hohen Strafen rechnen.

Vor allem sollte davon abgesehen werden, anderen Menschen Schaden zuzufuegen.

2.2 Hinweise

Am Rande sollte allerdings erwaehnt werden, dass das „versehentliche Sichten von nicht oder nur unzuellaenglich gesicherten Daten“ nicht strafbar ist. Im Falle eines geschwichten Netzwerkes mag das zutreffen, wer allerdings weitere Sicherheitsshuerden ueberwindet, begibt sich schnell auf sehr duennes Eis.

Auf was ich auch hinweisen will ist, dass das Ganze hier nicht aus meinen Wissen entstanden ist, sondern dass die Tools und das Know-How darueber schon seit langem existieren.

2.3 Grundlagen

Allgemeine Grundkenntnisse in UNIX, C++, TCP/IP und Netzwerktopologien sollten vorhanden sein.

3 Einfuehrung

Hier findet eine allgemeine Erklarung ueber Hijacking statt. Wer sich mit dem Grundprinzip von Hijacking bereits bechaefigt hat und wer weiss, wie TCP und aehnliche Protokolle funktionieren, kann diese Kapitel ueberspringen.

3.1 allgemein

Hijacking dient dazu bereits vorhandene Verbindungen zu uebernehmen. Hierbei kann es sich um Peer-to-Peer oder Client-Server Verbindungen handeln.

Juggernaut war hier eines der ersten Programme das sowas machte. In einem solchen Fall uebernimmt man entweder mittendrin die Verbindung zwischen zwei Rechnern oder man uebernimmt die Verbindung zu dem Zeitpunkt, an dem der Benutzer seine Verbindung abbaut. Wobei letztere Methode vorzuziehen ist, da der User davon nichts mitbekommt. Aber selbst bei einer Uebernahme waehrend der Sitzung kriegt der User, ausser eines simplen „connection reset by peer“ oder „connection close by remote host“ nicht viel mit. Darueber macht man sich auch weiterhin keine grossartigen Gedanken. Normalerweise.

Stoeren von Verbindungen dient dazu, normales Arbeiten unmoeglich zu machen. Das kann zum Beispiel genutzt werden, um zu verhindern, dass sich das Opfer nochmal einloggen kann und unsere Session, die wir zuvor gehijacket haben, schliesst. Auch kann damit der Zugriff auf eine Website gestoert werden, was man zum verfaelschen von Online-Abstimmungen oder Wahlen nutzen koennte. Ebenso ist es denkbar, damit eine Auktion oder Boersen-Seiten zu blockieren. Das erfordert aber mehr Aufwand, als eine sniffing Attacke. Hierbei sollte man eher auf das Attackieren von Routing-Protokollen sein Augenmerk legen, wozu man sich bei <http://www.phenoelit.de/> Anregungen holen kann.

3.2 Schichtwerk

Hijacking funktioniert auf verschiedenen Ebenen. Hier mal kurz erklart, was man bei den einzelnen Schichten so machen kann:

- Layer-2
Auf Layer 2 benutzt man ARP-Spoofing und ARP-Relaying Attacks. Damit taesucht man einfach MAC-Adressen aus oder vor. Das funktioniert quasi ueberall, hilft aber bei verschluesselten Sitzungen nicht viel weiter.
- Layer-3/4
Hier Hijacked man TCP, bzw. IP, man verbiegt einfach ein paar IP Adressen.
- Layer-5
Hier kompromittiert man Protokolle auf Application-Ebene (HTTP, FTP, POP3, ...)

3.3 Hardwarezeuch

Damit man ueberhaupt etwas sniffen kann, muessen ein paar Hardware und infrastrukturelle Vorraussetzungen gegeben sein:

- Der Angreifer mu in der Lage sein, Pakete fuer mindestens eines der Opfer empfangen zu koennen.
- Am besten ueber einen Hub verbunden, switches egal ob managebar oder nicht.
- Die Netzwerkkarte des Angreifers muss in den „promiscuous mode“ schaltbar sein (Realtek taugt da nix :-)).

4 Netztopologien

Es gibt noch ein paar mehr Netzwerktopologien als die hier vorgestellten, wir beschaeftigen uns aber hier nur mit zwei Arten von Netzwerken, wobei die zuerst vorgestellte heutzutage beinahe nicht mehr anzutreffen ist.

Warum ich hier was ueber Netzwerkstrukturen schreibe, hat den Hintergrund, dass es eigentlich immer so ist, dass irgendwas erklart wird, was eigentlich schon jeder weiss und schnell ueberflogen wird. Zumindestens ging's mir immer so beim Lesen mancher Artikel.

4.1 Shared Medium Netzwerke

4.1.1 Einfuehrung

Ein nicht geschwitchtes Netzwerk besteht immer aus einem shared Medium. Das heisst, alle Packete die ueber das Netz gehen, laufen schoen an allen Netzwerkkarten vorbei. Das sind beste Vorraussetzungen um zu sniffen. Es ist jederzeit moeglich, alle Verbindungen, die ueber das selbe shared Medium gehen, abzugreifen. Ich gehe hier nicht sonderlich darauf ein, weil das ganze einfach zu langweilig ist und sowieso jeder kann. Ausserdem ist diese Art der Netzwerkverkabelung am aussterben.

4.1.2 Beispiel

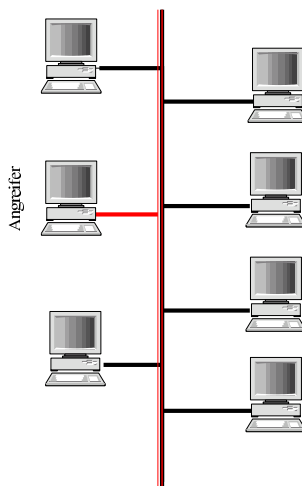


Abbildung 1: Aufbau eines shared Networks

4.2 geschwitchte Netzwerke

4.2.1 Einfuehrung

Ein geschwitchtes Netzwerk ist die heute gaengige Methode, Rechner untereinander zu vernetzen. Der Switch ist somit Hauptbestandteil eines jeden solchen Netzes. Er ermoeoglicht es, Datentransfers zwischen verschiedenen Usern zu unterhalten, ohne dass diese sich gegenseitig beeinflussen, bzw. sich gegenseitig Bandbreite nehmen. Dies ermoeoglicht es, Collusions zu vermeiden, bzw. auszuschalten.

Sobald ein Switch eingeschaltet wird und die angeschlossenen Geraete untereinander kommunizieren, legt der Switch eine Tabelle an, in der die MAC-Adresse steht und der dazugehoerige Port ueber denn diese erreichbar ist. Wenn nun ein Rechner A am Port 1

etwas an Rechner B an Port 2 sendet, so forwarded der Switch diese Packet, ohne dass die anderen Ports und somit Rechner etwas davon mitkriegen. Wenn nun A mit B und C mit D kommuniziert, so werden diese Verbindungen nicht durch Collusions gestoert und beide koennen mit der maximalen Bandbreite ihre Daten uebertragen.

Wenn ein Rechner ein Packet sendet, dessen MAC Adresse noch nicht in der internen MAC-Adressen-Tabelle ist (zum Beispiel wenn die Adresse hinter einem LAN switch liegt), so sendet der Switch dieses als Broadcast an alle Ports. Ausser natuerlich an den Port, von dem die Anfrage kam. Sonst wuerde schnell ein MAC Adressen-Flooding losgehen. Also immer wenn eine unbekannte MAC Adresse auftaucht, wird ein Broadcast gesendet und der Switch arbeitet in diesem Fall als Hub.

Manchmal liest man auch etwas von einer „transparent bridge“, das dient aber eher zur Verwirrung und wird nur von Marketingleuten oder unfaeihigen Dozenten¹ benutzt. Ist hier an dieser Stelle nur erwaeht, damit man das zuordnen kann, wenn man sowas mal hoert.

4.2.2 Beispiel

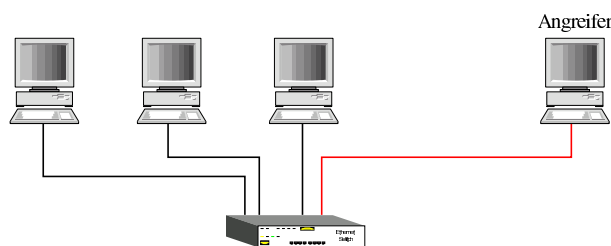


Abbildung 2: Aufbau eines switched Networks

5 Hijacking

Der Name Hijacking kommt ja eigentlich aus der USA und bedeutet soviel wie (Flugzeug-)Entfuhrung. Anders als allerdings wie bei einem Flugzeug merkt das Opfer allerdings nicht, dass es gerade entfuehrt worden ist. Normalerweise zumindestens, aber lassen wir das und folgen fort.

5.1 ARP Spoofing/Relaying - Layer 2

ARP Spoofing dient wie der Begriff schon sagt einem anderen irgendetwas vorzuspielen was gar nicht stimmt und damit Verwirrung zu stiften. Bei ARP Relaying heuchelt man zwar auch was falsches vor, allerdings mit dem Hintergedanken, sich als jemand anderes auszugeben. Auf diese Weise kann man relativ einfach Netzwerke manipulieren.

5.1.1 Was'n ARP?

Das Address Resolution Protocol (ARP) dient der Umsetzung von IP Adressen in Ethernet Adressen. ARP Anfragen werden als Broadcasts an alle Netzwerkteilnehmer versandt. Wenn keine Antwort vom gesuchten System kommt, senden alle Gateways die Suchmeldung an alle angeschlossenen Netzwerke weiter.

ARP dient zur physikalischen Adressierung des lokalen Netzes, so die generelle Umschreibung. Es dient auch dazu die Netzwerkadapter auf MAC-Ebene zu indentifizieren. Kennt man zum Beispiel die IP-Adresse wo ein Paket hinsoll, aber nicht die MAC-Adresse,

¹der betreffende Mensch an der FH-R ist wohl bekannt und darf sich auch ruhig angesprochen fuehlen

dann findet man anhand eines ARP-Request die passende MAC zur IP raus. Die eigene MAC-Adresse wird hierbei gleich immer mit uebermittelt. Das alles sollte aber soweit bereits aus jeder Netzwerkvorlesung oder irgendeiner Lektuere bekannt sein.

5.1.2 Kids-club

Der weniger spektakulaere Angriff ist eigentlich gar kein Angriff sondern dient lediglich dazu das Netzwerk zu ueberlasten. Hier nur ein kurzer Einblick, wie das prinzipiell moeglich ist, ohne weitere Ausfuehrungen. Das Stoeren von Netzwerken wird im allgemeinen als „lame“ abgetan. Im naechsten Punkt wird's dann wieder intressanter.

Die Suche nach nicht existenten IP Adressen mit kuenstlich erzeugten ARP Paketen erzeugt einen Broadcaststurm, der grosse Teile der Bandbreite belegt und den Netzbetrieb empfindlich stoert. Durch gefaelschte Antworten des nichtexistenten Systems, die wieder per Broadcast in alle Teilnetze verbreitet werden, kann die Last noch weiter erhoeht werden. Ziel dises Angriffes ist es, einen nicht definierten Zustand durch Ueberlastung des Netzes hervorrufen, im dem man einen Angriff starten kann. Durch Ueberlastung koennten (was eher nicht angenommen werden kann) Sicherheitsmechanismen ausfallen, so da der Schutz des internen Netzes verringert wird.

5.1.3 blonde Hardware

Viele billige switches, eigentlich alle, haben nur einen begrenzten Speicher, in den sie MAC-Adresse aufnehmen koennne. Typischerweise sind das 8000Adressen. Ist dieser MAC-Table-Overflow erreicht, wird die aelteste MAC-Adresse verworfen. Nun wird mit Hilfe eines Programms, wie zum Beispiel „macof“, ein Switch so geflooded, dass die MAC-Table staendig voll ist. Das hat zu Folge, dass bei jeder neuen MAC-Adresse die wirklich existiert, der Switch in den Broadcast-Modus umschaltet und das Packet an jedem Port rauslaesst. Somit verhaelt sich der Switch in diesem Fall wie ein Hub und „normales“ sniffen ist moeglich.

Ausserdem verhalten sich Low-End- und kleinere Office-Switches so, dass sie bei grosser Uebertragungslast in einen HUB-Modus schalten. Das passiert, wenn die Backplane die Daten nicht schnell genug verarbeiten kann, oder unter aehnlichen Umstaenden, die zu einer Ueberlast fuehren.

5.1.4 Gundel Gaukelei

Die eigene MAC-Adresse faelschen geht sehr einfach. Unter den meisten UNIX-System sollte das mit folgender Befehlsfolge funktionieren.

```
ifconfig <interface> down
ifconfig <interface> hw ether C0:01:CA:FE:BA:BE down
ifconfig <interface> 192.168.1.23
ifconfig <interface> up
```

Die Adresse kann man nun einfach per ICMP-echo-request (ping halt) an eine IP-Adresse (Rechner) oder per Broadcast an andere Rechner bekannt machen.

5.1.5 Spoofen

ARP Spoofing im nichtgeswitchten Netzwerk geht sehr einfach. Stellen wir uns mal folgendes Szenario vor:

Der Angreifer (192.168.1.23) generiert ein gefaelschtes ARP-Reply Packet und schickt dieses an das Opfer (192.168.1.42). In dem ARP-Paket steht nun folgendes drin.

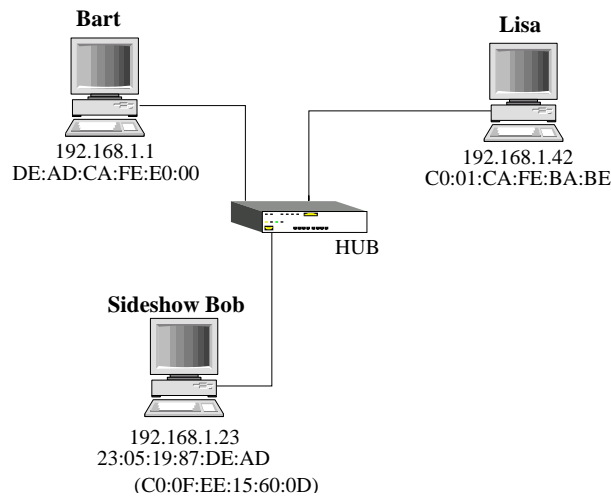


Abbildung 3: einfaches ARP Spoofing

```
C0:01:CA:FE:BA:BE arp reply 192.168.0.1 is-at C0:0F:EE:15:G0:0D
src-ip: 192.168.1.1
dst-ip: 192.168.1.42
```

Daraufhin aendert nun „Lisa“ ihre ARP-Tabelle und will mit „Bart“ gar nichts mehr zu tun haben, sondern schickt lieber alles an „Sideshow Bob“, was unserem Rechner entspricht. Der Rechner „Bart“ kriegt das ganze unter Umstaenden mit als Kernelmeldung

```
(C0:0F:EE:15:60:0D is using my ip
```

was aber unwahrscheinlich ist weil es kein ARP Broadcast war.

5.1.6 Eisszeiten und Einfriermoeglichkeiten

Schuetzen gegen solche ARP Spoofs kann man sich selber indem man sichere Protokolle verwendet so das nichts im Klartext uebers Netz geht. Ebenfalls gibt es bereits Patches fuer Linux die er ermoeöglichen die ARP-Table einzufrieren, aehnlich wie bei switches (siehe weiter unten).

Ein Befehl der die ARP Tabelle einfriert lautet:

```
arp -v -i eth0 -s 213.233.70.1 00:31:6B:94:32:A8
```

Sobald man dieses Kommando eingegeben hat kann man diesen ARP Eintrag weder loeschen, updaten oder faelschen. Ausser natuerlich man aendert ihn wieder manuell.

Soweit ich weiss kann man denn NetBSD und OpenBSD Kernel so compilieren das er ARP anfragen. die von sonderbaren IP Adressen kommen (z.B. 0.0.0.0 oder der eigenen), nicht reagiert. Das ist zwar noch nicht optimal aber besser als gar nichts.

5.2 TCP - Layer 4

Auf Layer 4 befindet sich TCP und UDP als gaengige Protokollfamilie.

Um die theoretischen Teil des Abschnittes zu verstehen werden Grundzuege des Protokollaufbaus von TCP [RFC 793] vorrausgesetzt.

TCP unterstuetzt eine full duplex Verbdingung zwischen zwei Punkten. Eine Verbindung ist eindeutig durch das „quadruple“ (IP Adresse des Senders, TCP Portnummer des Senders, IP Adresse des Empfaengers, TCP Portnummer des Empfaengers), definiert. Jedes

Byte, das hin- oder hergeschickt wird, ist durch eine 32 Bit grosse Sequenznummer markiert und wird mit einem Acknowledge des Empfängers bestätigt. Die Sequenznummer ist eine pseudo Zufallszahl und wird vor dem Verbindungsaufbau bestimmt.

5.2.1 Grundlegendes

Folgende Abkürzungen werden verwendet:

- SVR_SEQ: Die nächste Sequenznummer die vom Server gesendet wird.
- SVR_ACK: Das nächste Byte das vom Server empfangen werden sollte (Sequenznummer des letzten empfangenen Pakets plus eins).
- SVR_WIND: Empfangsfenster des Servers.
- CLT_SEQ: Die nächste Sequenznummer die vom Client gesendet wird.
- CLT_ACK: Das nächste Byte das vom Client empfangen werden sollte.
- CLT_WIND: Empfangsfenster des Clients.

Am Anfang einer Übertragung, wenn noch keine Daten gesendet wurden, gilt: $SVR_SEQ = CLT_ACK$ and $CLT_SEQ = SVR_ACK$. Dies gilt ebenso, wenn die Verbindung in einem „quiet“ Zustand ist (wenn auf beiden Seiten keine Daten gesendet werden). Sobald man etwas sendet, ändert sich dieser Zustand.

Generell kann man sagen gilt folgendes:

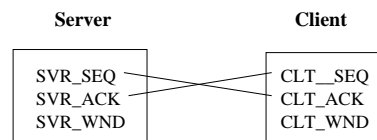


Abbildung 4: TCP Sequenznummernverhalten

$$\begin{aligned} CLT_ACK &\leq SVR_SEQ \leq CLT_ACK + CLT_WIND \\ SVR_ACK &\leq CLT_SEQ \leq SVR_ACK + SVR_WIND \end{aligned}$$

Zudem gibt es noch Control Bits:

- URG: Urgent Pointer
- ACK: Acknowledgment
- PSH: Push Function
- RST: Reset the connection
- SYN: Synchronize sequence numbers
- FIN: No more data from sender

Auf eine Übersetzung habe ich in diesem Fall mal verzichtet. Zudem brauchen wir nicht alle Flags, sie sind hier nur der Vollständigkeit halber aufgeführt.

Das Optionsfeld kann folgende Zustände beschreiben:

- SEG_SEQ: Bezieht sich auf die Paket Sequenznummer (wie wir im Header bereits gesehen haben).
- SEG_ACK: Bezieht sich auf die Paket Acknowledge Nummer.
- SEG_FLAG: Bezieht sich auf die Control Bits.

Diese sind hier erwähnt, weil sie für den Verbindungsaufbau nötig sind.

5.2.2 Verbindungsaufbau...

Ein typischer Verbindungsaufbau sieht wie folgt aus: Die SEQ_SEQ ist auf CLT_SEQ und SEG_ACK to CLT_ACK gesetzt. TCP benutzt einen „three-way handshake“ um eine Verbindung aufzubauen.

Hier mal ein typisches Beispiel:

Der Client sendet: $SEG_SEQ = CLT_SEQ0$
 $SEG_FLAG = SYN$

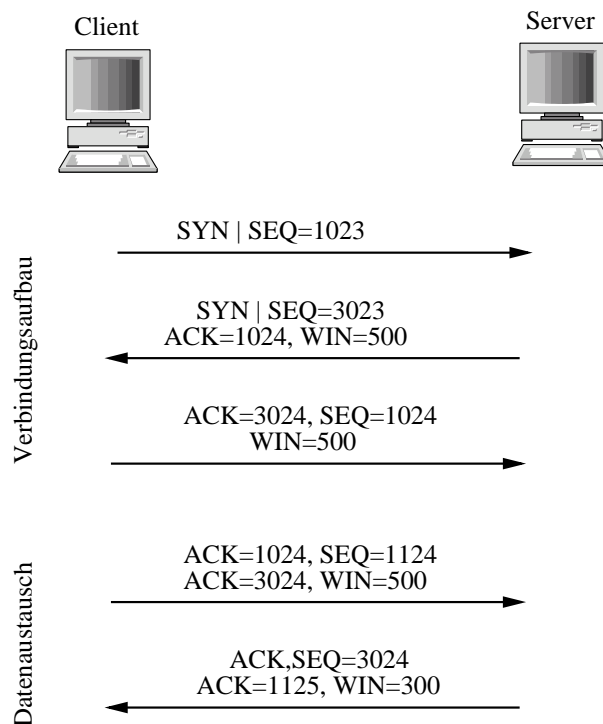


Abbildung 5: TCP Verbindungsaufbau, Client sendet 100 Bytes Daten

Ist dieser Handshake erfolgreich verlaufen steht die Verbindung auf TCP-Ebene und es werden anschliessen Daten uebertragen.

5.2.3 ... und Abbau

Wenn man nun alle seine Daten uebertragen hat, will man irgendwann mal die Verbindung auch wieder trennen. Dies geschieht bei TCP auf verschiedene Arten.

- Timeout
- Gleichzeitiger, beidseitiger Abbau mit FIN
- Einseitiger Abbau mit RST
- Rechner explodiert, Nuklearkrieg, etc ...

5.2.4 RFC 793, RST und andere Phaenomene

Die RFC schweigt sich ziemlich darueber aus, was RST betrifft. Auch weiterfuehrende und aktuellere Drafts legen sich nicht fest, was passiert wenn ein RST gesendet wird. Man sollte sich aber einen Ueberblick verschaffen und die betreffenden Seiten lesen. Das benutzen wir naemlich im folgenden fuer unseren Angriff.

Wann wird nun dieses RST Kommando gesendet?

- Antwort auf ein Packet einer nicht existierenden Verbindung.
- Antwort auf ein ACK eines noch nicht gesendeten Packets. ACK ist groesser als SEQ.
- Bei Verbindungsende einer Transmission.

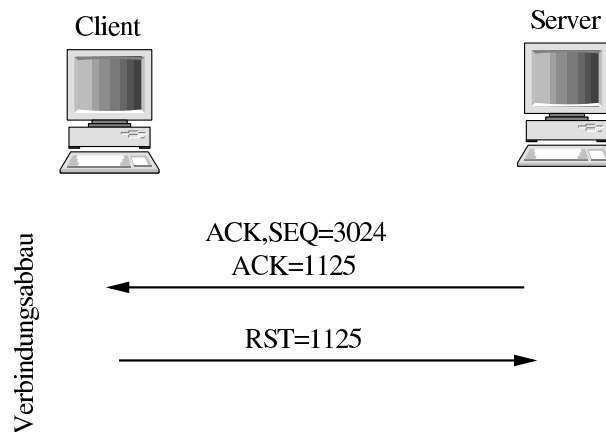


Abbildung 6: TCP Verbindungsabbau, Client sendet RST

- Verwerfen des Packets bei einer Firewall.

In dem Artikel von Laurent Joncheray „A Simple Active Attack Against TCP (1995)“, wird ein Hijack-Angriff mittels „RST-Reopen“ beschrieben.

Es berichtet darüber, dass sich Verbindungen nach einem RST wieder spontan aufbauen oder reconnecten. Das ganze ist allerdings stark von den implementierten Protokollstacks abhängig, klare Aussagen ueber Verhalten von Netzwerkverbindungen koennen daher nicht gemacht werden.

5.2.5 Desynchronized state

Die Verbinung befindet sich im „desynchronized state“ wenn sich die Verbindung in folgendem Zustand befindet.

- die Verbindung ist in einem ESTABLISHED mode
- ist in einem stable Zustand (keine Daten werden gesendet)
- die Server Sequenznummer ist **nicht** gleich der Client Acknowledgenummer ($SVR_SEQ \neq CLT_ACK$)
- die Client Sequenznummer ist **nicht** gleich der Server Acknowledgenummer ($CLT_SEQ \neq SVR_ACK$)

Falls Daten gesendet werden, koennen folgende zwei Sitationen eintreten:

1. Wenn die Konstellation so aussieht: $CLT_SEQ < SVR_ACK + SVR_WIND$ und $CLT_SEQ > SVR_ACK$ wird das Paket akzeptiert, die Daten werden fuer den spaeteren Gebrauch zwischengespeichert (abhaengig von der Stackimplementation) aber nicht an den User weitergegeben, da die Server Sequenznummer (SVR_ACK) fehlt.
2. Wenn $CLT_SEQ > SVR_ACK + SVR_WIND$ oder $CLT_SEQ < SVR_ACK$ gilt, ist das Packet ungueltig und wird verworfen. Die Daten gehen verloren.

Um eine Verbindung zu desynchronisieren, ist es am guenstigsten, die Server-Sequenznummer/Acknowledgenummer zu aendern. Es gibt zwei (moeglicherweise auch noch mehr) Moeglichkeiten dies zu erreichen:

Eine davon ist, ein Paket an der Server zu senden, welches keine Daten enthaelt, Voraussetzung dafuer ist aber, dass wir als der „eigentliche“ Client identifiziert werden. Damit erreicht man, dass die SVR_SEQ und die SVR_ACK des Servers sich erhoehrt.

Die andere Moeglichkeit ist, ein RST an den Server zu schicken. Somit bricht die Verbindung auf Server-Seite zusammen. (Ebenso koennte man ein FIN senden, was den gleichen Effekt hat. Allerdings wuerde dieses FIN von dem Server mit einem ACK bestaetigt werden, waehrend das RST nicht bestaetigt wird.)

5.2.6 Angriff

Vorraussetzung fuer den Angriff ist, dass IP-Spoofing funktioniert, so dass mit Hilfe von IP Adress Spoofing gefaelschte IP Pakete von aussen in das interne Datennetz gesendet werden koennen. Jedes Paket, das vom Angreifer geschickt wird, muss mit der IP und mit der MAC Adresse des Opfers manipuliert werden.

Der Angriff basiert nun darauf, auf beiden Enden einen desynchronisierten Zustand herzustellen, so dass beide Enden keine Daten mehr austauschen koennen. Der Angreifer konstruiert dann Pakete fuer die jeweils richtige Seite die so aussehen, als waeren sie die Pakete des eigentlich Ursprungs.

Gehen wir davon aus, dass die TCP Session in einem desynchronisierten Zustand ist und dass der Client ein Paket sendet.

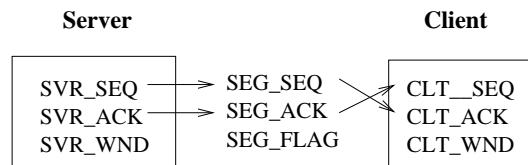


Abbildung 7: TCP Sequenznummern Pakete

SEG_SEQ = CLT_SEQ
SEG_ACK = CLT_ACK

Solange nun $CLT_SEQ \neq SVR_ACK$ gilt, werden alle Datenpakete verworfen. Der Angreifer sendet nun das gleiche Paket, aendert dabei aber die SEG_SEQ and SEG_ACK (und die Checksum) so dass folgendes gilt:

SEG_SEQ = SVR_ACK
SEG_ACK = SVR_SEQ

und der Server das Paket akzeptiert. Das Paket wird vom Server verarbeitet.

Falls $CLT_TO_SVR_OFFSET$ auf $SVR_ACK_CLT_SEQ$ **und** $SVR_TO_CLT_OFFSET$ auf $CLT_ACK_SVR_SEQ$ zeigt, ist der erste Teil der Attacke, dass der Angreifer die Pakete, die vom Client zum Server gehen, wie folgt umschreibt:

SEG_SEQ <= SEG_SEQ + CLT_TO_SVR_OFFSET
SEG_ACK <= SEG_ACK - SVR_TO_CLT_OFFSET

Der Angreifer kann somit jedes Paket, das zwischen zwei Punkten uebtragen wird, abfangen und somit wiederum jedes IP Paket ueber seinen Rechner leiten. Der Eindringling kann somit jegliche Art von Befehlen oder Daten in die Verbindung einfuegen, aendern oder loeschen. Bei einer Telnet-Session kann man zum Beispiel Befehle einfuegen und unerwuenschte Ausgaben weglassen. In diesem Fall muss man natuerlich die CLT_TO_SVR_OFFSET und SVR_TO_CLT_OFFSET aendern, bzw. anpassen.

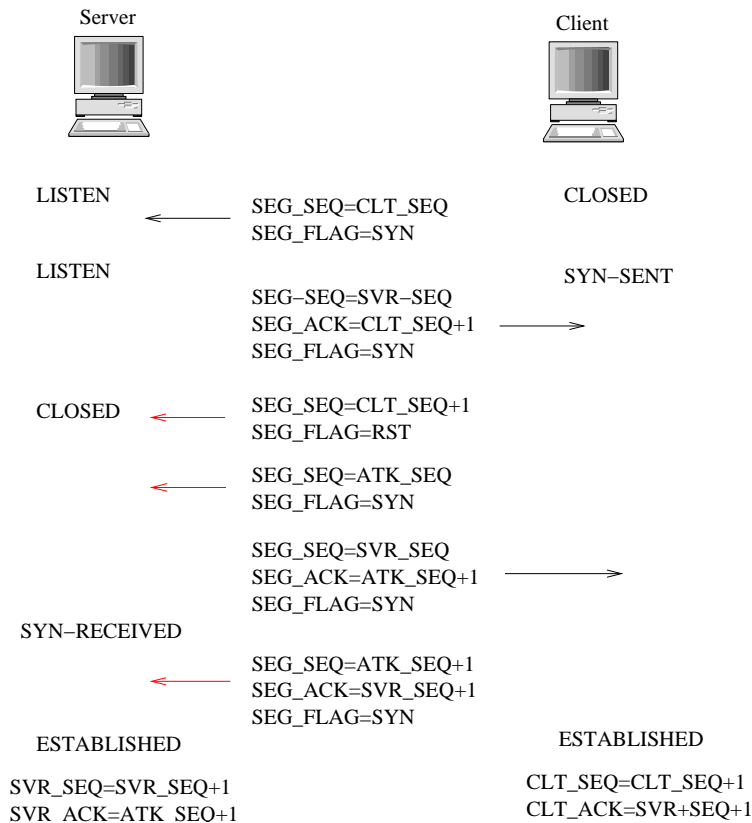


Abbildung 8: TCP Sequenznummern Attack

5.2.7 ACK-Storm

Ein grosser Nachteil dieser Hijacking Methode ist, dass ein ACK-Storm losbricht. Zwischen dem Server und dem ursprünglichen Client werden Pakete hin- und hergeschickt, die sich untereinander versuchen klar zu machen, dass seine Sequenz- bzw. Acknowledge-Nummer die richtige ist. Beide haben ja recht mit ihrer Behauptung und somit nimmt das Spiel seinen Lauf. Wie auch immer, diese ACK-Schleife läuft solange, bis entweder die Verbindung wegen Überlast zusammenbricht oder einer der IP-Stacks ist schlau genug, einmal aufzugeben und die Verbindung zu dropen. Zweiteres geschieht normalerweise in diesem Fall.

5.2.8 Kevin vs. Tsutomu

Nun ein kurzer Auszug aus dem „TCP Sequence Number Guessing“ Angriff von Kevin Mitnick auf die Workstation von Tsutomu Shimomura.

Die Wahl der Anfangssequenznummern erfolgt dabei dem äusseren Anschein nach zufällig, wird aber in Wirklichkeit aufgrund eines einfachen Algorithmus ermittelt. Laut RFC 693 sollte alle 4 Mikrosekunden der 32 Bit grosse Zähler der SEQ-Nummer um eins erhöht werden. In Wirklichkeit erfolgt die Erhöhung während einer Verbindung jede Sekunde um den Wert 128. Bei einem neuem Verbindungsaufbau wird der Wert um 64 erhöht. Mit diesem Wissen ist es möglich, mit hoher Wahrscheinlichkeit die nächste Sequenznummer beim Verbindungsaufbau zu erraten. Das gilt allerdings nur fuer Systeme, die vor mehreren Jahren eingesetzt wurden, die BSD Familie und Linux ab Kernel 2.4.x benutzen echte Zufallszahlen als Startwert und zaehlen keineswegs nur einfach hoch.

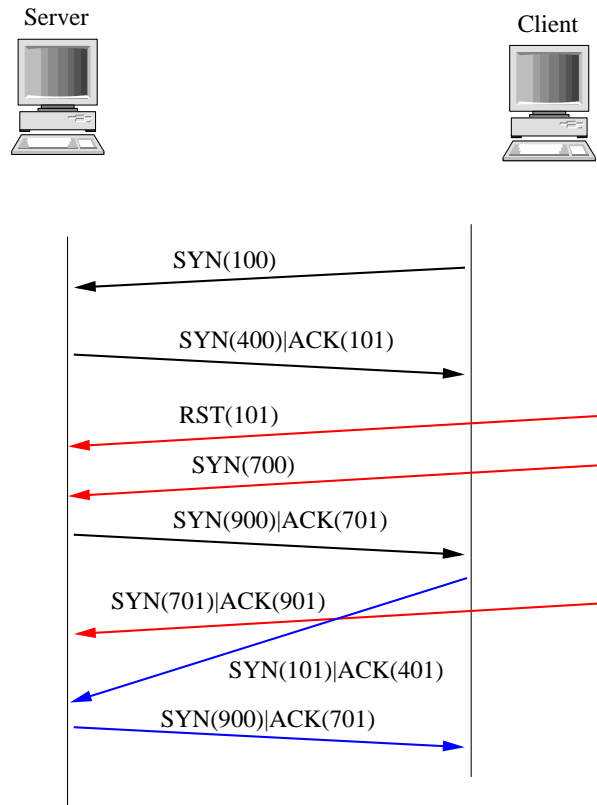


Abbildung 9: TCP ACK Storm, ausgelöst durch eine Hijack-Attacke

Selbst wenn der Angreifer die Anfangssequenznummer des Opfers nur ungefähr voraussagen kann, ist der Angriff immer noch vielversprechend. Der Angreifer sendet nach seinem TCP SYN Paket mit der gespooften Adresse immer eine ganze Reihe von IP Paketen mit gleichem Inhalt, aber unterschiedlicher Sequenznummer ab, so dass alle potentiellen Nummern abgedeckt werden. Alle Pakete bis auf das Richtige werden vom Opfer verworfen. Ziemlich praktisch fuer den Angreifer.

Mitnick versucht zunaechst mit den Befehlen finger, showmount und rpcinfo die Beschaffenheit des Systems Auszukundschaften:

```

14:09:32 toad.com# finger -l @target
14:10:21 toad.com# finger -l @server
14:10:50 toad.com# finger -l root@server
14:11:07 toad.com# finger -l @x-terminal
14:11:38 toad.com# showmount -e x-terminal
14:12:05 toad.com# finger -l root@x-terminal
  
```

Es folgt ein SYN Flooding Angriff auf Port 513 (Login/Port) von 130.92.6.97 zur Vorbereitung des nachfolgenden Angriffs. Damit wird der Login/Server ausgeschaltet, damit er bis auf weiteres keine neuen Verbindungsaufbauversuche akzeptieren kann:

```

14:18:22.516699 130.92.6.97.600 > server.login: S 1382726960:1382726960(0) win 4096
14:18:22.566069 130.92.6.97.601 > server.login: S 1382726961:1382726961(0) win 4096
14:18:22.744477 130.92.6.97.602 > server.login: S 1382726962:1382726962(0) win 4096
14:18:22.830111 130.92.6.97.603 > server.login: S 1382726963:1382726963(0) win 4096
????
????
  
```

```
14:18:25.483127 130.92.6.97.627 > server.login: S 1382726987:1382726987(0) win 4096
14:18:25.599582 130.92.6.97.628 > server.login: S 1382726988:1382726988(0) win 4096
14:18:25.653131 130.92.6.97.629 > server.login: S 1382726989:1382726989(0) win 4096
```

Nun sendet Mitnick (apollo.it.luc.edu) eine Testreihe von Verbindungsaufbauversuchen an x-terminal.shell um das Verhalten des Sequenznummerngenerators zu untersuchen. Die Sequenznummern (kursiv) erhhen sich bei jedem Verbindungsaufbauversuch um 128000:

```
14:18:34.452830 apollo.it.luc.edu.984 > x-terminal.shell: R 1382727007:1382727007(0) win
0
14:18:34.714996 apollo.it.luc.edu.983 > x-terminal.shell: R 1382727007:1382727007(0) win
4096
14:18:34.885071 x-terminal.shell > apollo.it.luc.edu.983: S 2024000000:2024000000(0) ack
1382727008 win 4096
14:18:34.962030 apollo.it.luc.edu.983 > x-terminal.shell: R 1382727008:1382727008(0) win
0
14:18:35.225869 apollo.it.luc.edu.982 > x-terminal.shell: S 1382727008:1382727008(0) win
4096
14:18:35.395723 x-terminal.shell > apollo.it.luc.edu.982: S 2024128000:2024128000(0) ack
1382727009 win 4096
14:18:35.427150 apollo.it.luc.edu.982 > x-terminal.shell: R 1382727009:1382727009(0) win
0
14:18:35.735077 apollo.it.luc.edu.981 > x-terminal.shell: S 1382727009:1382727009(0) win
4096
14:18:35.935681 x-terminal.shell > apollo.it.luc.edu.981: S 2024256000:2024256000(0) ack
1382727010 win 4096
```

Nun tauscht Mitnick Pakete des zuvor ausgeschalteten Login/Servers (Port 513) vor und sendet eine Verbindungsaufforderung an x - terminal:

```
14:18:36.245045 server.login > x-terminal.shell: S 1382727010:1382727010(0) win 4096
```

Aufgrund der zuvor getaetigten Beobachtung des Sequenznummerngenerators kann er die Sequenznummer (kursiv) des SYN/ACKs von server.login bestaetigen, ohne dieses Paket empfangen zu haben. Sie ist mit 1014384001 wieder genau 128000 hoeher als die SYN/ACK - Sequenznummer der letzten Testverbindung:

```
14:18:36.744422 server.login > x-terminal.shell: . ack 2024384001 win 4096
```

Damit hat Mitnick eine unidirektionale Verbindung mit x-terminal.shell aufgebaut, die von server.login zu kommen scheint.

5.2.9 Verteidigung

Wie verhindert man nun eine solche Attacke? Es gibt ein paar Moeglichkeiten, sich gegen solche Angriffe zu schuetzen, allerdings bieten diese keinen 100% Schutz, sondern vermindern das Risiko nur. Wenn man alle Sicherheitsmassnahmen beachtet, sollte man einigermaßen gegen Angriffen von aussen geschuetzt sein.

- Unterdrueckung von IP-Spoofing - also Inputfilterung (siehe weiter unten).
- Vermeidung der Authentifikation eines Benutzers auf der Basis von IP-Adressen. Firewallsysteme, die darauf aufbauen, sind als problematisch anzusehen.
- Die Authentifikation sollte nicht mittels der IP-Adresse, sondern durch ein Cryptosystem (Einmalpasswort etc.) geschehen.
- Implementierung eines echten Zufallsgenerators zur Berechnung der Anfangssequenznummern

Einige Sicherheitsunternehmen werden sicherlich auch vorschlagen, „kritische“ Seiten zu filtern. Davon ist generell Abstand zu nehmen. Ein Grund dafuer ist, dass es momentan keinen Filter gibt, der zuverlaessig solche Seiten sperrt, zumdem wechseln staendig die Namen, Inhalte, IP-Adressen, . . . , so dass eine generelle Filterung in keiner Weise moeglich ist. Desweiteren stellt eine Sperrung von Seiten eine Unterdrueckung der Meinungsfreiheit dar. Wer dieses Recht untergraebt, sei es auch (Anfangs) nur zu gutem Zwecke, wagt meistens auch den naechsten Schritt zu tun Wohin das fuehrt, sollte einem die Geschichte zeigen. Statt content Filterung lieber gleich das Surfen im Internet verbieten.

5.3 UDP - Layer 4

Manche beschimpfen UDP als „besseres IP Paket“. Damit haben sie im Grunde genommen auch Recht. RFC 768 beschreibt UDP als Verbindungsloses Protokoll. Es findet kein Verbindungsaufbau oder Abbau statt. Das muss die Anwendung die darueber liegt alles beruecksichtigen. Der Header eines UDP Pakets besteht nur aus Portnummer, der Laenge und der Checksumme. Somit ist das ganze sehr einfach aufgebaut und dementsprechen einfach zu spoofen.

Zu allem Ueberfluss bauen aber einige wichtige Dienste auf UDP auf. Zu diesen zaehlen unter anderem.

- NFS
- SNMP
- DNS
- Netbios
- RIP
- ...

Der Grund warum UDP eingefuehrt wurde liegt in der Idee begruendet das wenn man nur wenig Daten uebertragen muss nicht auch noch einen Verbdindungsaufbau und Abbau vornehmen muss. Bei einer DNS Anfrage zum Beispiel wuerde mehr Zeit benoetigt um eine Verbindung herzustellen als die eigentliche Abfrage dauert. Mit der Datenmenge verhaelt es sich aenlich. Fuer solche Anfrage ist es also angebracht auf UDP zu setzen.

Inzwischen gibt es bei TCP aber auch die Moeglichkeit beim Verbindungsaufbau schon Daten mitschicken. In RFC 1644 wird beschrieben wie man bei einem Verbindungsaufbau gleich Daten mitschicken kann und zugleich auch ein FIN, so dass die Verbindung auch wieder abgebaut wird. Das ermoeglicht eine fast gleichgute Uebertragungsleistung wie bei UDP aber mit etwas mehr Sicherheit. aktuelle UNIX Systeme unterstuetzen diese TCP extensions bereits. Nachteil dabei ist allerdings das beide Seiten dies beherrschen muessen.

Im Anschluss werden zwei moegliche DoS-Angriffe gezeigt die man mit UDP durchfuehren kann. Diese (D)DoS-Attacken hat nun aber nichts mit unserem sniffen im Netzwerk zu tun sondern koennte nur dazu verwendet werden das das Opfer fuer kurze Zeit ausgeschaltet wird. Solange man zum Beispiel seine geklaute Verbindung benutzt, oder aehnliches.

5.3.1 Flutwellen,...

Da UDP eine Flusskontrolle fehlt, kann es das Datennetz lahmlegen, indem es Router und Hosts mit Paketen ueberflutet. UDP haelt sich an dieselben Konventionen fuer Port Nummern und Server wie TCP, jedoch in einem eigenen Adressraum. Meist verwenden die Server niedrige Port Nummern. Da es keine virtuellen Verbindungen gibt, werden alle Pakete, die fuer einen bestimmten Zielport bestimmt sind, unabhengig von Quelladresse oder

Quellport an denselben Prozess weitergereicht. Aufgrund dieser Eigenschaft von UDP ist es moeglich beliebig viele Pakete an einen Rechner zu schicken. Diesem bleibt in dem Fall nichts anderes uebrig als zu versuchen alle zu verarbeiten. Dies fuehrt in den meisten Faellen zur voelligen Auslastung des Systems und der Bandbreite. Das intressante dabei ist das es sich dabei um ein voellig korrektes Paket handelt das eben von mehreren Rechner auf das Opfer geschickt wird.

5.3.2 Stuerme ...

Ein Broadcast Sturm kann man sich ebenfalls zu nutze machen. Bei diesem Angriff sendet der Angreifer sehr viele ICMP-Pakete (z.B. Ping-Anfragen) an die Broadcast-Adresse eines Netzwerks, so dass dieses Paket von jedem Rechner innerhalb des Netzwerks empfangen wird. Der Angreifer tarnt sich jedoch nicht mit seiner eigenen oder einer nicht-existenten Adresse, sondern mit der Adresse des eigentlichen Opfers. Die ICMP-Anfragen werden nun um die Anzahl der Rechner im Netzwerk vervielfacht; das Netzwerk dient quasi als Sprungbrett. Ist das Netz korrekt konfiguriert, werden die ICMP-Antworten am Router des Netzwerks abgeblockt und gelangen nicht bis zum Rechner des Opfers. Erlaubt das Netz jedoch solche ICMP-Broadcasts gegen aussen, werden die multiplizierten ICMP-Antworten an das Opfer weitergeleitet. Dadurch koennen Angreifer mit geringer Leitungskapazitaet (Modem, BA-ISDN) Opfer mit breitbandigen Anschlssen mit ICMP-Paketen ueberfluten. Die ICMP-Antworten belegen die gesamte Leitungskapazitaet und die regulaere Datenkommunikation wird unterbunden.

5.3.3 ...und andere Katastrophen

UDP-Pakete sind leichter zu faelschen als TCP-Pakete, weil es weder Quittungs- noch Laufnummern gibt. Es ist daher aeusserste Vorsicht geboten, wenn man die Quelladressen solcher Pakete zur Authentisierung verwendet. Die Applikationen selbst muessen geeignete Sicherheitsvorkehrungen treffen.

Hier mal kleine Beispiele was alles so passieren kann: Sendet man an einen Windows Rechner zum Beispiel ein Namensdienstdatagramm (netbios-ns 137//udp) mit einer nicht angeforderten negativen Antwort auf den Versuch einer Namensregistrierung fuer einen Namen, der lokal registriert ist passiert folgendes.

- der „Browserdienst“ verweigert ein Durchsuchen der Netzwerkumgebung.
- es werden keine Nachrichten mehr empfangen (net send), severmeldungen werden ignoriert.
- Domaenendienste koennen aufgrund der falschen authentiaet nicht mehr ausgefuehrt werden.
- freigegeben Ressourcen, sowohl auf dem eigenen Rechner sowie auch bezogene fallen aus.

Wie man sieht kann man nur durch ein einziges Paket einen Rechner ziemlich stark in seiner Arbeit behindern.

Bei NFS ist es sogar noch schlimmer. Hier werden alle Daten im Klartext uebertragen. Zusaetzlich verlaesst man sich auf die IP-Adresse als Authentifizierung. Das sind bei weitem nicht alle Luecken von NFS sondern nur die schlimmsten. Nicht umsonst wird es auch als „Nightmare File System“ von einigen Administratoren bezeichnet.

5.3.4 Deiche, Daemme und brennende Waende

Sich gegen solche Angriffe zu schuetzen ist in einigen Bereichen gar nicht moeglich. Viele der alten Dienste sind nicht darauf ausgelegt worden eine sichere Verbindung herzustellen.

len sondern eine schnelle. Es funktionierte damals und bei der Anzahl der ueberschaubaren Rechner waren Uebeltaeter schnell entlarvt. Heutzutage im beinahe (oder schon doch schon?) ausverkauften IPv4 Raum kommt es auf ein paar Bytes mehr die ueber die Leitungen wandern nicht mehr an.

Mit Firewalls kann man sich im allgemeinen gut gegen unerwuenschte UDP Pakete aus dem Internet erwehren. Von dieser Seite aus gehen nur noch wenige Gefahren aus die man aber nicht unterschuetzen sollte. Im internen Netz wo viele dieser Dienste ja ansetzen ist es allerdings viel schlimmer solche sachen in den Griff zu kriegen.

Wenn man alle nicht benoetigten Dienste abdreht und seine Router so konfiguriert das sie (zum Beispiel kein direct Broadcasting erlauben), sich nicht gerade Freundlich gegeneber UDP verhalten ist schon viel geholfen.

Zusaetzlich sollte man auf neuere Protokolle oder Dienste umschwenken falls dies moeglich ist.

5.4 ICMP - Layer 3

Das „Internet Control Message Protokoll“ fuer was es gut ist und was man damit anstellen kann.

5.4.1 RFC 792 und andere Zustaende

ICMP wurde erfunden um Aussagen ueber den Zustand eines Netzes machen zu koennen. Wie im vorrigen Kapitel gesehen haben befindet sich TCP auch in gewissen Zustaenden wie ESTABLISHED, SYN-SENT, LISTEN oder aehnlichen, das sind aber keine ICMP Zustaende sondern Zustaende einer Verbindung. Was man noch wissen muss ist das ICMP in einem IP-Packet verschickt wird, was aber eine triviale Feststellung ist.

Hier einige Beispiele:

- Echo/ Echo Reply (Ping)
- Destination Unreachable (Host down, Firewall)
- Source Quench
- Redirect
- Route Advertisement
- Information Request/Reply
- ...

Mit einiges dieser Befehle lassen sich Angriffe auf Netzwerke oder Rechner fahren. Es gibt sicherlich noch mehr Moeglichkeiten als die hier aufgefuehrten, aber diese sind die gaengigen Methoden um den normalen Betrieb zu stoeren oder sich Eigenheiten der Netzwerke zu nutzen zu machen. Die jeweiligen Auswirkungen sind trivial zu folgern und werden deswegen hier nicht weiter beschrieben.

5.4.2 Router Advertisement

Im normalfall bekommt ein IP-Host sein default-route manuell eingestellt, bzw. ueber DHCP was auch einer gewissen manuellen Einstellung genuege tut. Diese benutzt er als default gateway, alternativ kann man aber auch Router so konfigurieren das sie in periodischen Zeitabstaenden „ICMP Router Advertisement“ Pakete verschicken. Das heisst jetzt nicht das diese periodisch ausgesendeten Pakete ein auf ICMP basiertes routing protocol sein sollen. Das bedeutet nur das es moeglich ist auf passive Art und Weise seine nachstgelegenen verfuegbaren Routern zu erlernen.

Das Packet enthaelt fuer gewoehnlich die IP Adresse des Routers der die ICMP Router Advertisement Nachricht geschickt hat. Ebenso enthalten ist eine „Lifetime“ des wertes fuer wie lange der host den routingeintrag als aktuell ansehen soll bevor er ihn verwirft. Im normalfall sind das 30minuten.

Nachdem diese 30min vorbei sind, wird der eintrag aus der „routing table“ entfernt. Der Host bittet dann mit hilfe eines ICMP Router-Packets um eine neue Adresse oder er wartet weiterhin passiv auf ein ICMP Packet. Normalerweise wird alle 10min so ein Packet verschickt. Genaueres kann man in der RFC 1256 „ICMP Router Discovery Message“ nachlesen.

5.4.3 Redirection - Umleitung

Diese Nachricht wird von Routern benutzt, Hosts, die mit minimaler Routing - Information neu am Netzwerk aktiv werden (und denen z. B. nur die Adresse eines einzigen Routers bekannt ist), zur Benutzung der optimalen Route zu veranlassen. Router selbst sollten allerdings im allgemeinen so konfiguriert werden, dass sie immun gegen solche ICMP - Redirect - Nachrichten sind, und in jedem Fall ausschliesslich nach den Routern ihrer Routing - Tabellen vermitteln.

Angreifer von auerhalb sind sonst in der Lage, mit Hilfe von IP - Adress - Spoofing entsprechende ICMP - Redirect Nachrichten in das Netzwerk einzuschleusen und Router so zu einer nderung der Vermittlungswege ber beliebige Stationen zu bewegen. Gelingt es so, Verbindungen ber den externen Knoten des Angreifers selbst umzuleiten, ist ein massiver Netzeinbruch die unmittelbare Folge.

Ein grosser Vorteil gegenueber ARP-Spoofing ist das die Routen nicht nach einer gewissen Zeit ablaufen. Theoretisch kann somit auch eine Attacke von ueberall auf der Welt ausgefuehrt werden. Der Router leitet zudem die empfangenen Pakete trotzdem zum Ziel weiter, man bleibt also weitgehend unentdeckt.

5.4.4 Source Quench

Solche Art von Nachrichten werden nur von Gateways erzeugt. Der Hintergedanke dabei war das das Gateway eine gewisse Flusststeuerung uebernehmen kann. Kommen zum Beispiel sehr viele Pakete von einer Sendestation an so wuerden andere unter umstaenden benachteiligt weil das Gateway nur mit diesem einem Datenstrom beschaeftigt ist. Daher kann es sogenannte „Source Quench“ Pakete an den betreffenden Client schicken so das dieser seinen Durst nach Daten erst mal etwas zurueckstellt. Dieser Reduziert solange seine Datenuebertragung bis keines der SQ Pakete vom Server mehr ankommt. Durch dauerhaftes senden eines solchen Paketes kann die Datenuebertragung stark beeintraehtigt werden, so das der Client sich keine Pakete mehr verschicken traut.

5.4.5 Durstloescher und andere Schutzmassnahmen

Gegenmassnahmen zu den beschriebenen Angriffen sind nicht ohne weiteres zu treffen, da ICMP eine notwendige Teilkomponente des Internet Protokolls darstellt. Die leider gaengige Einstellung heutzutage mehr oder weniger alle ICMP Pakete an der Firewall zu dropen hilft zwar gegen solche Angriffe im Normalfall ist aber eher eine schlampige Methode sich dieser Angriffe zu erwaehren und zeigt eigentlich nur das der Betreffende Admin nicht recht viel Ahnung von der Materie hat. Zudem gibt es Angreifern die nuetzliche Information das es sich hierbei um eine Firewall handelt, da die Pakete verworfen werden und somit kein Timeout zurueckkommt. Also weniger geschickt das ganze.

Inzwischen kann man Router Systeme so zu konfigurieren, dass nur eine bestimmte maximale Anzahl von ICMP - Nachrichten pro Zeiteinheit in das interne Netzwerk vermittelt werden koennen. Im Normal Betrieb sollte die Anzahl der ICMP - Pakete verhaeltnismaessig gering ausfallen. Die meisten Netzwerk Management Systeme sind in der Lage,

die Anzahl der uebertragenen ICMP - Pakete zu ueberwachen und bei einem auergewoehnlichem Anstieg einen entsprechenden Alarm auszuloesen oder Gegenmassnahmen einzuleiten.

6 Realitaetsabgleich

Nachdem man nun verschiende Techniken und Praktiken kennengelernt hat, mal ein Szenario wie ein solcher Angriff in der Realitaet aussieht. Diesem Abschnitt ist ein gewisser paranoider foerdendes Element nicht abzustreiten. Die verwendeten Programme werden im naechsten Kapitel vorgestellt.

6.1 Szenario 1

7 Tools

Hier nun ein paar Programme die man benutzen kann oder auch nicht. Man kann die interessante, lehrreiche aber harte Tour waehlen oder die einfache, teilweise langweilige aber schnell zum Erfolg fuehrende. Spricht man versucht das ganze mit mehr oder wenig vorhandenen UNIX-Bordmitteln zu machen oder man nimmt einfach ein Tool her das einem die ganze Arbeit abnimmt.

7.1 do it yourself

Es gibt hier soviele Programme das man schon blind sein muss wenn man keines findet. Zudem sollte jeder mal schon tcpdump in der Hand gehabt haben, also duerfte das keiner grossen Erklaerungen beduerfen.

Man braucht einen sniffer

- tcpdump
- snifit
- ethereal

einen Packetgenerator

- ippacket
- tkipconstructor
- spak

und noch paar Flooder

- ARP-Flooder
- SYN-Flooder
- RST-Deamon

alternativ kann man sich auch noch selber was mit libnet oder libpcap

7.2 ready, steady, hijack

Hier gibt es nicht (noch nicht) so viele Programme. Hier wiederum nur die wichtigsten.

- juggernaut (für geschaltete Netzwerke nicht so geeignet)
- hunt (stürzt öfter mal ab)
- dsniff (sehr nett, mit Plugins Unterstützung)
- ettercap (einfach zu bedienen und funktionell)

8 Schutzmassnahmen

8.1 Einführung

Nachdem das Problem das nun bekannt ist macht man sich natürlich Gedanken was man dagegen tun kann.

- IPsec verwenden
- verschlüsselte Methode oder Zertifikate verwenden.
- weniger anfällige Netzwerktopologien aufbauen.

8.2 Hardware

In neuen managbaren Switches gibt es die Möglichkeit den MAC-Adressen Table einzufrieren. Alle zu diesem Zeitpunkt bekannten Adressen werden gespeichert und eine Änderung dieser ist nicht mehr möglich. Ebenso ist es nicht möglich neue hinzuzufügen. Somit kann man verhindern das ein unbekannter einfach mitsniff in dem er ARP Spoofing betreibt. Wenn der allerdings vorher schon seine MAC-Adresse auf eine gültige Umgestellt hat hilft einem das auch nichts. Zudem ist dieser Einsatz nur denkbar wenn man ein starres Netzwerk besitzt. Wenn Mitarbeiter nur temporär mit Laptops oder ähnlichen elektronischen Geräten anwesend sind ist diese Schutzmassnahme verständlicherweise ungeeignet. Von diesen flexiblen Geräten geht aber meistens die Grösste Bedrohung aus.

Firewalls helfen hier nur bedingt. Ich sehe schon in Zukunft firewalls in Switches eingebaut damit man sich besser schützen kann. Die Hersteller lassen sich da sicherlicher ein paar passende Werbeslogans einfallen. Naja, firewalls helfen hier nur bedingt bis gar nicht.

Momentan gibt es keine mir bekannte Hardwarelösung die es ermöglicht festzustellen ob ein ARP-Sturm gerade losbricht, oder ob sich eine Netzwerkkarte im 'schnueffelnodus' befindet. Vielleicht hat ja einer eine gute Idee und das ändert sich in Zukunft.

8.3 Software

Wenn im Netzwerk ein ARP-Sturm losbricht sollte man auf alle Fälle was unternehmen. Das kann zwar auch eine defekte Netzwerkkarte verursachen, aber die müsste man ja eh suchen und austauschen. Um einen ARP-Sturm festzustellen gibt es bereits Software.

Ebenfalls kann man auch herausfinden ob sich eine Karte im Promiscuous Mode befindet, allerdings nur unter gewissen Voraussetzungen. Anhand der Reaktionszeit der Karte auf einen ping könnte man theoretisch feststellen ob sich die Antwortzeit verändert hat. Karten die sich im sniffing-modus befinden brauchen länger um zu reagieren. Das ganze hat aber bei ausgelasteten Netzwerken, oder stark schwankenden Netzwerklasten eigentlich keine Chance eine sichere Diagnose durchzuführen. Hier dürfte es zu häufigen Fehleralarmen kommen, die wiederum zu einem lapidaren handhaben der Angriffe führen. Somit gibt es keine sichere Methode festzustellen ob man abgehört wird oder nicht.

8.4 IP Filter

Hier gibt es sowohl Hardware wie auch Software Loesungen. Die sicherste Methode im Bereich der Softwareloesung duerfte wohl der Einsatz von OpenBSD mit ipf sein. Allerdings ist die Einrichtung und Administration kein Kinderspiel. Dafuer kann man sich danach ruehmen das man es nicht nur installiert hat sondern auch am laufen hat.

8.4.1 Inputfilterung

Im groben und ganzen gibt es gibt sechs Hauptgruppen von Absender-IPs, die Sie auf Ihrem externen Interface auf jeden Fall abweisen sollten. Das betrifft alle eingehenden Pakete, die angeblich von folgenden IP-Adressen ausgehen:

- Die IP-Adressen von einem internen Netzwerken wird niemals als eingehendes Packet ankommen. Kommt eines dieser Pakete an ihrer Firewall an sollte man diese sofort verwerfen. Die Adressen fuer den privaten IP-Adressraum sind in RFC 1918 definiert.
- Multicast-Adressen der Klasse D. IP-Adressen der Klasse D sind fr den Gebrauch als Empfaenger-IPs bei Multicast-Broadcasts reserviert, z.B. bei Audio- und Video-Uebertragungen. Sie liegen im Bereich von 224.0.0.0 bis 239.255.255.255. Diese Adressen sollten auf dem Internet nicht als Absender vorkommen.
- Oft vergessen werden die Reservierten Adressen der Klasse E. IP-Adressen der Klasse E sind zukuenftigen und experimentellen Anwendungen vorbehalten und werden nicht oeffentlich vergeben. Sie liegen im Bereich von 240.0.0.0 bis 247.255.255.255. Die NSA und das amerikanische militaer verwendet teilweise diese IP's, also besser gleich wegwerfen bevor sie was zu Gesicht bekommen was sie spaeter noch bereuen werden. Sie sollten niemals ein Paket von
- Das gute alte Loopback. Das Loopback-Interface ist ein privates Netzwerkinterface, das von Unix fr lokale netzwerkbaasierte Dienste eingesetzt wird, wie zum Beispiel X-Server aufrufe. Statt lokalen Netzwerkverkehr durch die Treiber des Netzwerkinterfaces zu schicken, nimmt das Betriebssystem das Loopback-Device und schont somit die eigentlich Netzwerkkarte. Der Loopback-Bereich geht uebrigens von 127.0.0.0 bis 127.255.255.255 und bezieht sich nicht auf eine Adresse wie faelschlicherweise immer angenommen.
- Broadcast auf das gesamte Netz. Ein Broadcast auf die Adresse 0.0.0.0 ist im Internet nicht zulaessig. Das wird nur in internen Netzen fuer DHCP oder BOOTP anfragen benutzt. Als externe Adresse sollte man diese wegwerfen.
- Auch eine beliebte Angriffsmethode, weil sie am haeufigsten vergessen wird sich dagegen zu schuetzen, ist die eigene IP-Adresse. Sie werden niemals eine Paket bekommen in dem Ihre IP Adresse steht. Also die eigene IP-Adresse sperren.

8.4.2 Empfaenger IP Filterung

Endlich gibts mal wenig zu tun. Bei der Empfaenger IP gibts deshalb wenig zu tun weil die Netzwerkkarte, bzw. die Protokollimplementierung sich eh nur fuer seine IP Adresse interessiert. Alles andere schaut sie gar nicht an. Jetzt kommt das aber. Aber Broadcast Adressen interessieren dagegen schon wieder. Die Broadcast Adresse 255.255.255.255 ist die allgemeine Adresse auf die alle reagieren muessen/sollten. Diese dient in manchen Faellen bei internen Netzwerken fuer "keep-alive" Meldungen. Ein Broadcast an die Empfaenger-Adresse 0.0.0.0 muss man anders verstehen. Es wird wie eine offizielle Adresse gehandhabt die von einer Broadcast-Adresse geschickt wurde. Ein Broadcast an die Absender-Adresse

0.0.0.0 statt an die richtige 255.255.255.255 richtet sich immer gegen eine UNIX Maschine. Aus historischen Gruenden reagieren implementierungen basierend auf BSD auf die Anfrage mit einer ICMP Antwort vom Typ 3. Alle anderen Betriebssysteme ignorieren diese Pakete. (Ich konnte allerdings diese Verhalten bei einem FreeBSD 4.4 nicht mehr feststellen)

Dies ist eines der wenigen Beispiel bei denen man ein Paket gleich verwerfen sollte (DENY) anstatt es als Fehlermeldung zurueckzuschicken (REJECT).

8.4.3 AbsenderPort Filterung

Hier gibt es nicht viel zu erklaren. Das es sich bei dem Client wieder Name schon sagt um jemanden handelt der gerne Daten austauschen wuerde wird er sich mit einem Server verbinden. Da die Serverports unter 1023 liegen wird der Client einen Port zwischen 1034 und 65535 benutzen. Falls das nicht so ist, sollte man das Packet verwerfen. Wie gesagt, nix spannendes.

8.4.4 EmpfaengerPort Filterung

analog zu AbsenderPort Filterung nur umgekehrt. langweilig.

8.4.5 TCP Flags

Die Regeln, welche eingehende Pakete angenommen werden sollen, knnen auch die Flags bercksichtigen, mit denen der Zustand der TCP-Verbindung angezeigt wird. Alle TCP-Verbindungen benutzen die gleiche Gruppe mglicher Verbindungszustnde. Wegen des dreiteiligen Handshakes beim Verbindungsaufbau unterscheiden sich die Flags bei Clients und Servern voneinander.

Wenn ein fremder Client eine Verbindung aufbaut, ist im ersten Paket des Handshakes das SYN-Flag gesetzt, das ACK-Flag aber nicht. Bei allen Paketen, die danach ankommen, ist nur noch das ACK-Flag gesetzt. Die Firewall-Regeln fr einen Server sollten alle eingehenden Pakete zulassen, unabhngig davon, welchen Wert SYN oder ACK haben.

Von einem fremden Server ankommende Pakete sind immer Antworten auf einen Verbindungsaufbauwunsch, den Ihr Client-Programm initiiert hat. bei jedem Paket ist also das ACK-Flag gesetzt. Firewall-Regeln fr einen Client sollten Pakete vom Server daher nur mit gesetztem ACK-Flag zulassen. Ein normaler Server wird normalerweise nicht selbst eine Verbindung zum Client aufbauen.

8.5 Umgang mit ICMP

Hier scheiden sich die Geister. War man frueher der Ansicht alle ICMP Pakete einfach wegzuschmeissen geht man seit neuerem einen anderen Weg. Es gibt aber immer noch ein paar Gruende ein Paket zu droppen.

- Die meisten Pakete werden verworfen, weil sie boesartig sind, nicht weil jemand ganz unschuldig auf einen Dienst zugreifen wollte, den man eben nicht anbietet.
- Jedes Paket auf das man antwortet kann als Teil eines Denial-of-Service-Angriffes (DoS) eingesetzt werden.
- Jede Antwort, selbst eine Fehlermeldung, liefert dem Angreifer potenziell nuetzliche Informationen.
- Eine Fehlermeldung verdoppelt den Verkehr auf dem Netz

Wobei letztere Grund eher ein schlechter Grund ist da ja ICMP dafuer gedacht und konzipiert wurde.

Man kanns nahezu nie richtig machen aber sicherlich falsch, so meine Meinung zu dem Thema. Deswegen kann man generell nur eine Empfehlung aussprechen die in diesem Fall meiner persoenlichen Auffassung entspricht.

- kein oder nur teilweises Versenden von ICMP Echo-Replys.
 - Spionagemoeglichkeit fuer interne Netzstrukturen.
- verhindern von ICMP Echo
 - Ping Flooding
 - nur zu wenigen Rechnern erlauben. Suchmaschinen nutzen dies fuer Webserver.
- kein versenden von „Destination Unreachable“
 - Auskunft ueber offene UDP Ports
 - Siehe nmap UDP scan oder nessus.
- ignorieren von ICMP Source-Quench
 - DoS durch einschleusen von ICMP Pakten mit diesem Inhalt. Erreichbarkeit sinkt.
- ignorieren von ICMP Redirect
 - DoS angriff oder Ueberlast
 - Man-in-the-Middle Attack
 - unerwuenschte IP Routen (traffic abkommen)
- verhindern von ICMP Time-Exceeded
 - Auskunft ueber interne Router und Bandbreiten (traceroute)
 - Firewall sollte nicht im Traceroute sichtbar sein.

Ein normaler gesunder Menschenverstand und ein ausreichendes Wissen ueber Netzwerke sollte es ermoeöglichen eine vernueftige sichere Loesung fuer eine ICMP Filterung hervorzubringen.

8.6 misstrauen

ein gewisses Mass an misstrauen schadet nie. Wenn einem die ssh Verbindung abraucht und man sich wieder einloggt und noch seinen 'alten' User sieht stoert einem das im generellen nicht. Die normale Einstellung ist 'der fliegt schon mal raus' - oder auch nicht. Wenn man feststellt das die idle Zeit dann auch ploetzlich sinkt in der 'verlorenen' Sitzung sollte man schleunigst dran gehen ein paar Prozesse zu killen.

Sorgsames lesen von SSL-Zertifikaten oder aehnlichen sollte man gewissenhaft machen (beware of Snake Oil!) und bei einer Aenderung drauf achten was man tut. Notfalls kann man mal ein paar cents investieren und bei der Firma anrufen ob sich denn was an ihrem Zertifikat geaendert hat. Vorausgesetzt man erwischt dort einen der sich damit auskennt.

Somit ist man auch fuer zukuenftige Gefahren besser geschuetzt als wenn man sich auf Hard-, oder Software verlaesst.

8.7 wenns doch passiert?

Einen Notfallplan vorher erstellen. Wenn einer mal eingeloggt ist heisst es schnell handeln. Der Angreifer kann schnell passwoerter Wechseln so dass man keine Chance hat den Rechner runterzufahren oder abzuschotten. Hier kann man nur noch den Stecker ziehen, nicht dem am Stromnetz sondern denn am switch. Eine kurzfristiges umschalten einer firewall auf 'deny from all' sollte auch helfen das der angreifer keine Chance mehr hat etwas anzurichten. Vorrausgesetzt man ist noch Herr ueber seine firewall.

Ein solches Szenario sollte man vorher mal durchspielen und vielleicht sogar mal ueben. Im Ernstfall zaehlt jede Sekunde und wenn man nicht genau weiss was man tut macht man unter Umstaenden mehr kaputt als das es hilft.

Einen generellen Ratschlag kann man hier nicht geben. Man sollte abwaegen was wichtiger ist. Ein Serverausfall von ein paar Stunden weil man den Stecker zieht oder ein paar tausend Kreditkartenummern von Kunden in falschen Haenden.

9 Quellen

http://www.Deter.com/unix/papers/tcp_attack.ps.gz

http://www.cs.purdue.edu/homes/clay/papers/tcp-ip/Simple_Active_Attack_Against_TCP.ps

10 Danksagung

- Zuerst mal den Jungs vom CCC. Die immer gute Arbeit leisten und immer wieder neues finden. Allen voran Stefan Krecher.
- Die verstaendnisvollen Admins der FH-Regensburg.
- Allen meinen Freunden die mich ermutigt und unterstuezt haben diesen Artikel zu schreiben.