Willkommen zur ersten Ausgabe von Spaxid !

:: Inhalt

:: Mitwirkende Autoren

Jonas Havers (jh@it-helpnet.de)

Nico 'Triplex' Spicher (triplex@it-helpnet.de)

Simon 'McSchlumpf' K. (mcschlumpf@it-helpnet.de)

Simon 'Zodiac' Moser (s.moser@wireless-bern.ch)

Simon 'echox' Koelsch (echox@echox.de)

Peter Wilfahrt (peter@antioffline.de)

Gerrit 'Notro' E. (notro@it-helpnet.de)

Christoph 'debu' Wille (debu1@firemail.de)

Thomas 'RoCKDaFrogg' Rogg (thomas@outcast-media.com)

(nach Artikeln und Artikeleinreichung geordnet)

:: Lektoren

Jonas Havers (jh@it-helpnet.de)

:: Allgemeiner Kontakt

E-Mail: mail@spaxid.it-helpnet.de

Website: http://spaxid.it-helpnet.de

Das Risiko dynamischer Webseiten: SQL Injection

Wohin der Trend führt

Heutzutage geht der Trend bei der Webseitenerstellung in Richtung dynamischer Webseiten, die sich leicht über Weboberflächen verwalten lassen, also so genannte Content Management Systeme (CMS). Neben diesen Systemen bedienen sich auch Webshops und Messageboards der Datenbankfunktionalität. Informationen, die auf Webseiten angezeigt werden, werden also in Datenbanken gespeichert und beim Webseitenzugriff abgerufen. Neben Oracle, FileMaker, IBM DB2, Access und diversen weiteren Datenbanken werden vor allem SQL-Datenbanken immer beliebter. Aus diesem Grund setzen Angreifer sehr häufig auf Fehler in solchen CM Systemen, um auf Informationen in den Datenbanken zuzugreifen, auf die sie normalerweise keinen Zugriff hätten, wie zum Beispiel Passwörter, E-Mail Adressen und Anschriften.

Funktionsweise dynamischer Webseiten

Dynamische Webseiten werden mittels Nutzung von Websprachen wie PHP, Perl, JSP, ASP u.a. erzeugt. Dabei werden Informationen aus Datenbanken oder Dateien ausgelesen und an den passenden Stellen auf der Webseite formatiert oder unformatiert ausgegeben. Zu diesem Zweck verwendet man die Befehlssprache SQL (Structured Query Language). Um die Ausgaben dem Besucher anzupassen, integriert man in die SQL-Anfrage die Benutzereingaben, die meist über ein Formular an die Webapplikation weitergegeben werden. Die Hauptbefehle für ein SQL Query sind SELECT, UPDATE, DELETE, INSERT, die zum Auswählen (SELECT), Erneuern (UPDATE), Löschen (DELETE) und Einfügen (INSERT) von Datensätzen dienen und recht leicht zu verwenden sind.

Definition von SQL Injection

SQL Injection ist aus dem weiter oben beschriebenen Grund eine sehr beliebte Angriffstechnik auf den Inhalt dynamisch-generierter Webseiten und zählt neben Cross Site Scripting mittlerweile zu den Top-Angriffsarten auf Webapplikationen. Sie wird oft fälschlicher Weise zum Cross Site Scripting (XSS) hinzugezählt. Richtig ist jedoch, dass sie sich etwa zwischen XSS und CGI-Vulnerabilities eingliedern lässt. Es handelt sich bei SQL Injection präzise gesagt um die Auslesung, Manipulation oder Löschung von Datensätzen in SQL Datenbanken, die durch eine Manipulation einer SQL Anfrage (SQL Query) durchgeführt werden kann. Das Hindernis hierbei ist jedoch, dass man die Tabellenstruktur kennen muss, um gezielte Manipulation an dem Tabelleninhalt vornehmen zu können.

Praxisbeispiele

Nach etwas Theorie wollen wir nun mit der Praxis fortfahren. Wie oben beschrieben werden Datenbankanfragen mit der Sprache SQL getätigt. Eine Abfrage sieht beispielsweise so aus:

```
SELECT * FROM user WHERE user_id = '123'
```

Diese Anfrage liefert alle Datensätze aus einer Datenbank mit der Tabelle *user* zurück, die in der Spalte *user_id* den Wert *123* aufweisen.

Um nun dieses Query individuell den Benutzereingaben anzupassen verwendet man Variablen, die ihren Inhalt über ein Formular erhalten. So sieht das SQL-Query mit Ersetzung des Wertes *123* gegen eine Variable mit Einbeziehung von PHP so aus:

```
SELECT * FROM user WHERE user_id = '$user_id'
```

Anmerkung: In der PHP-Version 4.1.0 sollte $user_id durch $HTTP_POST_VARS ersetzt werden; in allen neueren Versionen (> 4.1.0) durch $_POST[user_id] oder $_REQUEST[user_id].

Diese Anfrage kann nun von einem böswilligen Benutzer modifiziert werden. Gibt der Benutzer in das entsprechende Formularfeld nämlich beispielsweise *';DROP TABLE user--* ein, so sieht das SQL-Query folgendermaßen aus:

```
SELECT * FROM user WHERE user_id = '';DROP TABLE user--'
```

Die Erklärung ist schnell getan. Das Semikolon dient hierbei nämlich als Trennzeichen zwischen den zwei Befehlsketten SELECT... und DROP..., die Datenbank sieht somit zwei Befehle, die sie ausführt:

```
SELECT * FROM user WHERE user_id = ''
```

... zeigt alle Datensätze aus der Datenbank, deren Spalte *user_id* leer ist.

```
DROP TABLE user--'
```

... löscht die Tabelle *user* komplett.

Die zwei Bindestriche des zweiten Befehls dienen zur Kenzeichnung eines Kommentars, wodurch das letzte Hochkomma (engl.: Quote) ignoriert wird und die Anfrage somit keinen Fehler zurückliefert.

Ein weiteres Beispiel zeigt die Wirkungsweise von SQL Injection ebenfalls sehr deutlich. Gibt der Benutzer *123' OR 1=1--* ein, so werden alle Datensätze aus der Tabelle *user* und der Spalte *user_id* selektiert. Das Query sähe damit so aus:

```
SELECT * FROM user WHERE user_id = '123' OR 1=1--'
```

Diese Bedingung trifft immer zu, da 1=1 immer korrekt ist, und somit werden alle Datensätze aus der Spalte *user_id* zurückgegeben.

Ein letztes Beispiel, was ich hier noch anführen möchte, zeigt die Problematik nochmals. Wenn der Benutzer *123' UNION SELECT password FROM user_info--* eingibt, sieht das Query so aus:

```
SELECT * FROM user WHERE user_id = '123' UNION SELECT password FROM user_info--'
```

Durch den UNION SELECT Befehl kann man zwei separate SQL SELECT Anfragen durchführen. Dieses Query würde somit die Datensätze in der Tabelle *user*, Spalte *user_id* mit dem Wert *123* zurückliefern sowie die Datensätze der Tabelle *user_info* in der Spalte *password*.

Die Ausmaße und Möglichkeiten dieser Angriffstechnik sind Ihnen somit hoffentlich nun bewusst. Sie sind auf jeden Fall ernst zu nehmen, da ein Angreifer auf einfachste Art und Weise ganze Datentabellen aus Ihrer Datenbank ins Jenseits befördern und/oder sensible Daten aus diesen entnehmen kann.

Aktuelle Beispiele

PHP-Nuke 6.9 SQL Injection Vulnerability

Vor einiger Zeit (genau am 01.02.2004) wurde eine SQL Injection Schwachstelle von den Betreibern des Sicherheitsportals „Security Corporation" in dem weltweit verbreiteten Content Management System PHP-Nuke (Version 6.9) entdeckt. Durch diese Schwachstelle war und ist es möglich, Passwörter aller angemeldeten Benutzer (in verschlüsselter Form) sowie deren Informationen zu erhalten, die sie in ihrem Benutzerprofil gespeichert haben.
Quelle: http://www.security-corporation.com/advisories-027.html

Microsoft SQL Server SQL Injection Vulnerability

Diese Schwachstelle wurden von Marc und David Lichtfield vom NGSSoftware Sicherheitsteam gefunden und entsteht dadurch, dass eine Applikation die Benutzerangaben nicht richtig überprüft und voreilig in ein SQL Query einbindet. Ein Angreifer kann dadurch seinen eigenen Code dem Server übergeben und als unterpriviligierter Benutzer beliebige Befehle auf dem Server ausführen (sofern die Datenbank auf einem höherpriviligierten Benutzerkonto läuft).
Quelle: http://www.kb.cert.org/vuls/id/508387 und http://www.securityfocus.com/bid/5309

Ein kleines Beispiel meinerseits hierzu zum Abschluss. Gibt der Benutzer *123';EXEC master..xp_cmdshell dir--* ein, so wird in diesem Fall eine Liste der Dateien und Ordner im derzeitigen Ordner aufgeführt (durch den Befehl *dir*). Die Folgen dürften beim Betrachten der vielfältigen Möglichkeiten, die sich hieraus ergeben, ersichtlich sein (Erstellung neuer Benutzerkonten, Modifizieren oder Löschen von Dateien uvm.).

Schutzmaßnahmen

1. Sicheres Programmieren

Das erste, was Sie tun sollten, ist, sich Ihren Code einmal genauer anzuschauen und solche Vorkommnisse aufzudecken. Ein MySQL-Query, wie wir es oben behandelt haben, kann mittels PHP zum Beispiel mit der Funktion *addslashes* relativ gut gesichert werden:

```
mysql_query(„SELECT * FROM user WHERE user_id = '".addslashes($_POST['user_id'])."'");
```

Jedem Zitierzeichen wird hierdurch ein Backslash vorangestellt und somit wird die Anfrage im Falle eines Missbrauchs falsch und liefert einen Fehler zurück. Sofern magic_quotes_gpc in der php.ini aktiviert ist, wird die Funktion *addslashes* auch automatisch bei Benutzereingaben angewandt. Eine weitere Funktion wäre *mysql_escape_string*, die zwar etwas sicherer ist, jedoch in der Praxis unnötig ist, sofern *addslashes* verwendet wird.

2. Überprüfen eingegebener Zeichenketten

Sie sollten auf jeden Fall die vom Benutzer eingegebenen Zeichenketten auf deren Inhalt überprüfen und eventuell bösartigen Code herausfiltern. Hierbei sind die Eingabe von SQL-Befehlen und Sonderzeichen wie Quotes ('') oder Bindestrichen (--) vor allem zu beachten. Sie sollten also genau definieren, welche Eingaben zulässig und sinnvoll sind.

3. Sicherheitsmodul mod_security (Apache)

Für den Apache Webserver besteht ein zusätzliches Modul, welches Ihnen hilft, sich gegen Cross Site Scripting (XSS) und SQL Injection zu schützen. Mehr dazu auf www.modsecurity.org

4. Database-Intrusion-Protection-Systeme
Wer vollkommen sicher gehen möchte, der verwendet Database-Intrusion-Protection-Systeme (DB-IPS). Diese erkennen Angriffe auf Datenbanken und wehren diese ab. Durch gewisse Richtlinien sind sie zusätzlich in der Lage zu unterscheiden, welche Applikation und welche Benutzer bestimmte Befehle an die Datenbank senden dürfen. Einige dieser Systeme sind sogar fähig, Anomalien zu erkennen, d.h. im Fall, dass eine Applikation Befehle ausführen will, die vorher noch nie ausgeführt wurden, aber gültig sind, werden diese speziell protokolliert.

5. Lassen Sie sich und Ihren Code prüfen
Getreu dem Motto „4 Augen sehen mehr als 2" empfehle ich Ihnen auch hier vorzugehen. Geben Sie Ihren Code einem (oder mehreren) fähigen Programmierer, der sich ebenfalls mit diesem Thema schon auseinander gesetzt hat und derselben Sprache fähig ist, in der Sie Ihren Code verfasst haben. Eventuell werden so auch weitere Fehler oder Schwachstellen aufgedeckt und können bereinigt werden. Es existiert außerdem Software, die Ihre Webseite und/oder den Hosting-Server Ihrer Seite selbstständig auf SQL Injection Fehler und andere Fehler überprüft. Einige Beispiele im Folgenden:

Nessus Security Scanner
Nessus ist ein bekannter und äusserst wirkungsvoller Remote Security Scanner *(remote, engl.: entfernt).* Es existieren Plugins, mit denen man einen entfernten Rechner neben vielen Angriffstechniken auch auf SQL Injection prüfen kann.
Homepage: www.nessus.org

kses Security Scanner
kses ist eine Software, die Ihren Code auf potentielle Schwachstellen gegen Cross Site Scripting (XSS), Buffer Overflows, DoS Attacken und SQL Injection überprüft.
Homepage: http://sourceforge.net/projects/kses

Mieliekoek.pl
Hinter diesem Namen versteckt sich ein Perlscript von roelof@sensepost.com, welches Ihre Webseite auf SQL Injection hin überprüft.
Homepage: http://www.securityfocus.com/archive/101/257713

ATK - Attack Tool Kit
ATK ist ein kleines, handliches Security Auditing Tool von Marc Ruef, für das unter anderem auch einige Plugins zum Testen auf SQL Injection geschrieben wurden.
Homepage: http://www.computec.ch/projekte/atk/

Webgoat
Webgoat ist zwar kein Security Scanner, er vermittelt jedoch Entwicklern die Gefahren und Schwachstellen von webbasierten Applikationen anhand von interaktiven praktischen Beispielen. Unter anderem wird hier auch SQL Injection behandelt.
Homepage: http://www.owasp.org/development/webgoat

Schlusswort

Wie Sie sehen ist SQL Injection eine durchaus ernst zu nehmende Angriffstechnik, die immer wieder zum Einsatz kommen kann, da sich viele Code-Schreiber dieser Schwachstelle nicht bewusst sind oder sie einfach vernachlässigen. Machen Sie es besser und überprüfen Sie Ihren Code ! Ich hoffe, ich konnte Ihnen dazu einige interessante Informationen bieten. Falls weitere Fragen bestehen, so bin ich gerne bereit, diese per eMail oder im Forum auf www.computer-support.org zu beantworten.

-- Jonas Havers

Wireless Fidelity ?

Denial-of-Service-Angriffe gegen Wireless-LAN

Inhalt:

Einleitung

In den letzten Monaten sind die Verkaufszahlen von Wireless-LAN-Geräten stark gestiegen. Auch in vielen Privathaushalten und Firmen haben sich mittlerweile die drahtlosen Netzwerke gegenüber den oft unpraktischen und schwierig zu verlegenden Kabeln profiliert. Wie es zu erwarten war, sind auch diverse Sicherheitslücken, z.B. in WEP, dem gängigem Verschlüsselungs-Algorithmus, aufgetaucht. Da es aber genug deutschprachige Texte über die allgemeine (Un)sicherheit von Funknetzen gibt, wird sich dieser Text nur mit Denial-of-Service-Attacken (kurz: DoS) beschäftigen. Diese stellen besonders bei Einrichtungen, wie z.B. Krankenhäusern oder Bankinstituten ein sehr hohes Risiko dar.

Grundlegendes Wissen über Netzwerke, Sicherheit und W-LAN sollte vorhanden sein, um den Text zu verstehen.

DoS-Attacken

Zu Deutsch heißt Denial of Service etwa "Verweigerung der Dienste". Im IT-Bereich bedeutet das einen Computer oder ganze Netzwerke zum Erliegen zu bringen. Das Ziel eines solchen Angriffs auf ein WLAN ist, den Datenaustausch zu stören, und so User davon abzuhalten, auf das Netzwerk zuzugreifen. WLAN DoS-Attacken werden in zwei Unterarten unterteilt: Fehler im 802.11-Protokoll und das Attackieren des "Physical Layers" mittels Störsendern.

Schwachstellen im 802.11-Protokoll:

(Dies sind nur einige aller Schwachstellen, ich werde nur die gängigsten und gefährlichsten näher erklären.)

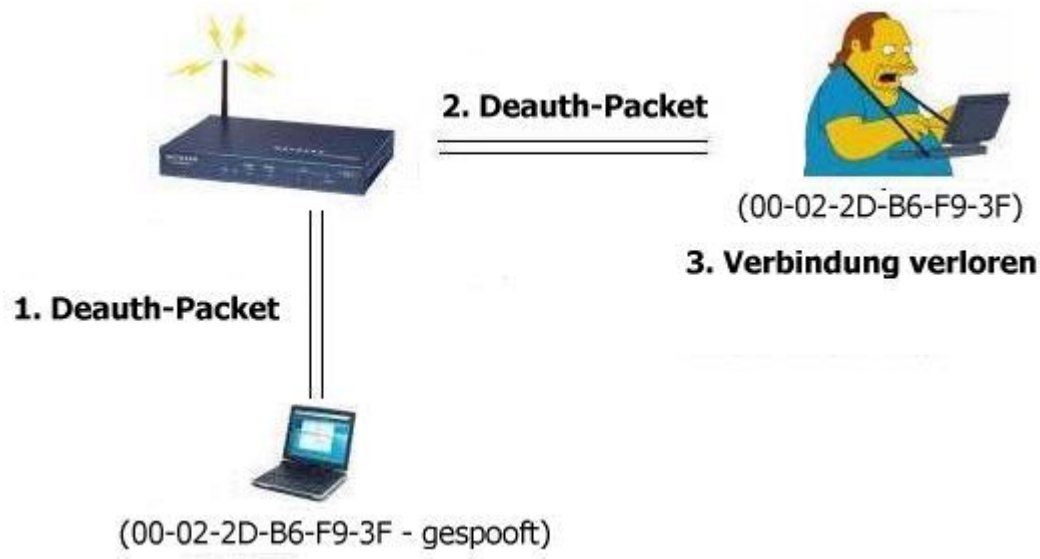
- **Deauthentication Flooding**

  Beim Deauthentication Flooding nutzt der Angreifer eine Schwachstelle im 802.11-Protokoll aus. Also ist bei diesem Angriff auch kaum Schutz durch einen Router o.ä. möglich.

  Normalerweise werden mit Deauthentication-Packeten am AP angemeldete Clients wieder abgemeldet. Die Schwachstelle besteht nun darin, dass nur an der MAC-Adresse geprüft wird, ob das empfangene Signal auch von dem richtigen Client kommt. Diese Lücke könnte man folgendermaßen ausnutzen:

  1. Der Angreifer spooft die MAC-Adresse eines bereits angemeldeten Clienten mit einem Tool wie smac [1]
  2. Er sendet die Deauth-Frames an den AP mit einem Tool wie void11 [2] oder Air-Jack [3]
  3. Der AP beendet die Verbindung mit dem richtigen Clienten, ebenfalls mit einem Deauth-Packet

  Folge: Der Client trennt, wie im Protokoll vorgesehen, die Verbindung und sucht nach einem neuen AP.



  Natürlich kann der Angreifer auch die MAC-Adresse des APs spoofen und das Deauth-Packet direkt an den Clienten schicken.

  Interessant ist, dass diese Angriffsmethode auch bei mit WEP verschlüsselten Netzen funktionert, da diese Daten (Deauth-Frames o.ä.) nicht verschlüsselt werden. Also ist es auch ein Leichtes mit einem herkömmlichen Sniffer wie Ethereal [4] an die benötigten Daten zu kommen.

  Außerdem gibt es auch eine ähnliche Attacke, gegen die nur APs anfällig sind. So wie mit Deauth-Packeten Verbindungen beendet werden, so baut man sie mit Association- und Authenticationpacketen wieder auf. So könnte man den Router auch mit hunderten simulierten Verbindungsanfragen flooden - jedoch sind die meisten APs gegen Angriffe solcher Art geschützt.

- **"Michael"**

  Michael ist ein in WPA, dem Nachfolger von WEP, integrierter "message integrity check"-Algorithmus - der eigentlich zum Schutz vor Angreifern dienen sollte. Jedoch bietet dieser ein theoretisches Einfallstor für DoS-Attacken: Michael deaktiviert den AP für einige Zeit, falls er per vermeintlicher Brute-Force-Attacke angegriffen wird. Sobald innerhalb einer Sekunde mehr als ein falscher Schlüssel an den AP geschickt wird, so werden alle Verbindungen für eine Minute oder länger getrennt. Diese Schwachstelle

könnte man nun aber ausnutzen und den Router auf Dauer immer wieder deaktivieren. Besonders kritisch an dieser Schwachstelle ist, dass der Angreifer nur nur sehr wenige Packete verschicken muss. Ein verstecker Handheld mit einem automatisiertem Script, versteckt in der Nähe des APs - jeder Systemadministrator dürfte sich nun über die Folgen bewusst sein.

- **DSSS CCA Algorithmus-Schwachstelle**

  Die folgende DoS-Attacke wurde erst kürzlich (17.05.2004) von dem US-CERT [5] entdeckt. Betroffen sind alle IEEE 802.11, 802.11b und langsame (weniger als 20 Mbit/s) 802.11g WLAN-Geräte. Sie zeichnet sich dadurch aus, dass sie sehr effektiv und schwer zu lokalisieren ist. Die Schwachstelle hängt mit der "medium access control" (kurz: MAC) zusammen. Im Falle von WLAN wird von allen Geräten "Carrier Sense Multiple Access with Collision Avoidance" (CSMA/CA) eingesetzt, um zu vermeiden, dass zwei Geräte gleichzeitig senden. Die Clear Channel Assessment (CCA) Prozedur, die im physischen Layer ausgeführt wird, ist essentiell um festzustellen, ob die Radiofrequenz frei zum Übetragen von Daten ist.

  Bei einem Angriff nutzt der Angreifer nun die CCA-Funktion aus um allen WLAN-Geräten in der Reichweite ein "volles" Netzwerk zu simulieren. Wärend des Angriffs werden so alle WLAN-Geräte keine Daten mehr schicken - dabei spielt es keine Rolle, ob Access Points oder Clienten angegriffen werden.

  Bisher ist, soweit ich zum aktuellen Zeitpunkt beurteilen kann, (zum Glück) noch kein Exploit o.ä. im Internet zu finden.

Jamming

Bei herkömmlichen Kabelnetzwerken kommt es manchmal zu einem Datencrash. Dieses passiert relativ selten, da auf ein Kabel nur wenige Faktoren einwirken (hohe Hitze, Kälte, usw). Wenn die Daten über Funk übertragen werden, ist die Lage jedoch weitaus schwieriger, da nun viel mehr Faktoren eine Rolle spielen. Unter "Jamming" versteht man das Attackieren des Physical Layers, bei WLAN den Frequenzbereiche 2,4 Ghz und 5 Ghz (bei 802.11a), mit einem Störsender.

Da die 2,4 Ghz "ISM"-Frequenz kostenlos zu nutzen ist, ist es nicht schwierig, Geräte zu finden, die in diesem Frequenzbereich senden. Bluetooth, Mikrowellen, Babyphones und viele andere Geräte senden auf dem ISM-Band - die Daten werden vermischt und der Datenaustausch kommt zum Erliegen.

Auch lassen sich bereits Störsender (siehe Bild) käuflich erwerben.

Zum Schluss

Wie man eindeutig sieht, ist Wireless-LAN bei weitem nicht so sicher wie die Industrie es den Käufern versichern will. Bisher sind Denial-of-Service-Angriffe zwar noch nicht besonders populär geworden, d.h. aber nicht, dass sie nicht möglich nicht - durch simple Methoden ist es für einen Profi möglich, ganze Netze zum Erliegen zu bringen. Weiterhin nimmt der Erfolg von WLAN aber nicht ab, im Gegenteil: immer mehr Benutzer steigen auf die drahtlosen Netzwerke um, immer neuere Standards erscheinend und die Bandbreiten werden immer höher.

Noch viele Fehler müssen behoben werden - doch alles was uns im Moment übrig bleibt, ist warten und hoffen, dass sich 802.11 zu einem sicheren und qualitativ hochwertigem Standard entwickelt.

Quellen

- http://www.auscert.org.au/render.html?it=4091
- http://www.arubanetworks.com/products/whitepapers/secure-wireless/index.php?pg=1
- http://opensores.thebunker.net/pub/mirrors/blackhat/presentations/bh-usa-02/baird-lynn/bh-us-02-lynn-802.11attack.ppt
- http://www.heise.de/security/
- http://www.cc.gatech.edu/classes/AY2004/ cs6255_fall/papers/Wireless-MITM-Attack-102803.ppt
- http://standards.ieee.org/getieee802/download/802.11-1999.pdf

Links

[1] http://www.klcconsulting.net/smac/
[2] http://www.wlsec.net/void11/
[3] http://802.11ninja.net/airjack/
[4] http://www.ethereal.com/
[5] http://www.us-cert.gov/

-- Nico 'Triplex' Spicher

"Wie funktioniert eigentlich eine Desktop Firewall...?"

Inhalt:

1. Beschreibung einer Desktop Firewall

Unter Desktop Firewalls versteht man auf einzelnen Workstations installierte Firewalls, die sehr eng mit dem Betriebssystem zusammenarbeiten. Dies hat den Vorteil, dass sie unerlaubten Zugriff auf das Internet, durch auf dem Computer installierte Programme, direkt blockieren können.

Die enge Zusammenarbeit mit dem installierten Betriebssystem hat allerdings den Nachteil, dass die installierte Desktop Firewall mit dem Betriebsystem kompatibel sein muss, was zum Beispiel bei einem Apple Macintosh schwierig ist. Außerdem kann die Firewall leicht durch Fehler wie Buffer Overflows ihre Funktion verlieren und somit dem Angreifer freien Weg gewähren, was bei einer Hardware Firewall allerdings ebenso der Fall ist.

Die Desktop Firewall verwendet meist die Urform und leichteste Variante der Firewalls, die sogenannten "**Paket-Filter**", die in Zusammenarbeit mit anderen Funktionen verwendet werden. Ein Paket-Filter filtert, wie ein Wasserfilter, die unerlaubten Programme oder Tätigkeiten heraus und stoppt diese. Dabei werden die Header der ankommenden Datenpakete auf die Protokolltypen IP, ICMP, TCP und UDP geprüft und mit den vom Administrator konfigurierten Regeln verglichen (mehr hierzu unter Punkt 2).

Viele Desktop Firewalls verfügen zusätzlich noch über Zusatzfunktionen wie Werbeblocker, Kindersicherung oder auch einem integrierten Virenschutz. Nicht nur aus diesem Grund verwendet man in vielen kleinen Netzwerken Desktop Firewalls als einzigsten Schutz, indem man sie auf den einzelnen Workstations installiert, wobei dies für größere Netzwerke allerdings abzuraten ist. Da es zum Einen viel Aufwand und Zeit kostet, für jeden einzelnen Computer das entsprechende Regelwerk zu definieren, und zum Anderen es so für den Angreifer ein leichtes ist, durch die Überwindung einer Desktop Firewall, Zugriff auf das gesamte Netzwerk zu erlangen.

2. Technik des Paketfilters

Wie schon unter Punkt 1 beschrieben verwendet ein Paketfilter vordefinierte Regeln, die von dem zuständigen Administrator angelegt wurden. Diese werden bei dem schnellen Blick auf die ankommenden Header der Datenpakte mit meist folgenden Kriterien verglichen:

- Quell-IP-Adresse
- Ziel-IP-Adresse
- Quell-Port
- Ziel-Port

Im Regelwerk können nicht nur einzelne "Quell-IP-Adressen" blockiert werden, sondern auch ganze Netzwerkadressen. Außerdem unterstützt der Paket-Filter nicht nur die Filterung von

Paketadressen, sondern auch die von Ports, wovon man dann von dem sogenannten "Portfiltern" spricht. Hierbei geht der Paket-Filter allerdings nur von "Well-Know-Ports" aus wie zum Beispiel das dem Port 21 zugeordnete FTP. Da Ports auf den Schichten 3 und 4 im OSI-Schichtenmodell angesiedelt sind, überwacht ein Paketfilter also keinen Datenverkehr auf höherer Ebene, da sich ein Paketfilter voll und ganz auf die Header von Paketen konzentriert und keine Informationen wie Verschlüsselungen oder Benutzernamen aus den Headern von Paketen der Protokolltypen IP, ICMP, TCP und UDP erkennbar sind.

Da der TCP-Header Angaben über Portadressen enthält, kann so der Paket-Filter feststellen, welcher Dienst und somit welchen Zweck der Datenverkehr hat; der eigentliche Inhalt des Pakets bleibt jedoch ungewiss. Einige Paket-Filter entscheiden auch noch nach Kriterien wie zum Beispiel die Anzahl der gleichzeitigen Verbindungen, letzte Anmeldung des Administrators, der Firewall oder Uhrzeit und Wochentag - diese Kriterien zählen allerdings nicht zum Standardumfang von Desktop Firewalls.

Etwas weiter als ein reiner Paketfilter gehen Session Level Firewalls. Sie konzentrieren sich nicht nur auf die Header von Paketen, sondern auf ganze Sequenzen von Paketen.
Eine Session Level Firewall kann so Verbindungsaufbau und -abbau überprüfen und ungültige Datenpakete verwerfen. So können Scan- und wiederholte Verbindungsversuche oder auch Versuche von IP-Hijacking unterbunden werden.

Stateful Packet Inspection (SPI) erweitert Session Level Firewalls um einige Funktionen. Die Technik funktioniert anhand einer "Zustandstabelle" (state = Zustand), womit die Firewall entscheiden kann, was genau mit dem Datenpaket passieren soll. Anhand der gespeicherten Informationen in der oben genannten Tabelle kann und muss die Firewall entscheiden, wie mit dem Paket umzugehen ist (Erlauben? - Verwerfen? - oder einfach zurück an den Absender?).

Die Zustände unterscheiden sich allerdings immer von dem gerade verwendeten Protokoll (z.B. TCP oder SMTP) einer Verbindung zwischen zwei Verbindungspartnern, und ob das verwendete Protokoll zustandsorientiert ist oder nicht. Ist dies nicht der Fall, kann eine SPI logischerweise nicht stattfinden, da es keine Zustände gibt, die es zu analysieren gilt.

Bei dem Protokolltyp TCP beispielsweise gibt es viele verschiedene Zustände:
(...kleiner Ausschnitt...)
- CLOSED: Zustand nach einer Verbindungsbeendigung
- LISTEN: Zustand, wenn ein Dienst auf eine Verbindung wartet
- SYN_SENT: Zustand eines Hosts, wenn er eine Verbindungsanfrage gesendet hat
- SYN_RCVD: Zustand eines Hosts, wenn er soeben eine Verbindungsanfrage erhalten hat, und nun eine Bestätigung (SYN_ACK) schickt.
- ESTABLISHED: Zustand bei dem "Start-Host", der auf die ankommende Bestätigung (SYN_ACK) ebenfalls eine solche zu dem Verbindungspartner sendet.

Mit Hilfe dieser gespeicherten Zustände kann eine Firewall Attacken wie z.B. SYN Flooding, IP-Fragmentierungsattacken oder auch IP-Spoofing erkennen, was von großem Vorteil ist.

So kann der Angreifer nicht, oder nur erschwert, Zugriff auf das System erlangen, was bei einer erfolgreichen IP-Spoofing Attacke ein leichtes wäre. Ebenfalls werden Paket-Filter anstatt nur bei Desktop-Firewalls auch bei einigen Proxyservern oder auch Transportschicht-Gateways eingesetzt. Dies dient zum Schutz des Proxyservers und zum Schutz des internen Systems, zu dem evtl. offene Verbindungen bestehen.

2.1 Regelwerk von Paket-Filtern

Das Regelwerk eines Paket-Filters enthält folgende Regeln, die von Desktop Firewall zu Desktop Firewall gleichbleibend ist:
- Accept/Allow-Regeln lassen den Datenverkehr bzw. die erwünschten Pakete passieren
- Deny-Regeln verwerfen ein Datenpaket
- Reject-Regeln weisen das Paket an den Absender zurück, der eine Fehlermeldung erhält

Die Grundeinstellung bei den meisten Paket-Filtern, im Gegensatz zu selbstlernenden D. Firewalls, ist die "Deny-Regel", was die Folge hat, dass alle Paket verworfen werden, und somit keine Verbindungen enstehen kann. Das heisst, dass Sie erst einmal den erwünschten Datenverkehr, also die "Allow-Regeln", definieren müssen, sowie gegebenenfalls die "Reject-Regeln".

Bei einer selbstlernenden Desktop Firewall dagegen gibt es keine "Standard Deny-Regel". Sie werden bei jedem Datenverkehr ohne vorherig definierte Regel durch ein Dialogfenster gefragt, ob dieser Datenverkehr erwünscht oder unerwünscht ist.

Die meisten Desktop Firewalls gehen streng nach der Regel- oder Prioritätsliste von oben nach unten durch, was man unbedingt bei dem Definieren von Regeln beachten sollte. Dies hat den Nachteil, dass die meisten Paket-Filter auf diese Weise einfach die Überprüfung der Regeln auf ein ankommendes Datenpaket abbrechen, sobald sie eine passende Regel gefunden haben. Liegt nun z.B. eine Allow-Regel für Port XX vor einer Deny-Regel für eine bestimmte IP-Adresse in dem Regelwerk, so kann auch die eigentlich gesperrte IP-Adresse auf Port XX zugreifen.
Dies bedeutet, dass man für jegliche Anwendung eine Regel erstellen muss, damit die Benutzung eines Paket-Filters einen Sinn ergibt.

3. Funktionen der Desktop Firewall

Die Funktionen einer Desktop Firewall sind von Produkt zu Produkt sehr verschieden.
Die Bandbreite der Funktionen sind meist von dem Preis des Produktes abhängig und reichen von *Kinderschutz* bis hin zum *Werbeblocker*.

Die Kinderschutz-Funktion ist hauptsächlich für den Erziehungsberechtigten eine große Hilfe, damit Jugendliche oder Kinder keinen Zugriff auf nicht-jugendfreie Seiten haben.
Mit das der Fall ist, werden im Voraus bestimmte Seiten im Filter definiert, auf denen ein bestimmter Benutzer, hierbei z.B. das 11-jährige Kind, keinen Zugriff haben soll. Einige Desktop-Firewalls (z.B. Norton Internet Security) bieten auch vordefinierten Seiten bzw. Kategorien an, auf denen ein User keinen Zugriff haben sollte. Im Grunde genommen ist die Kinderschutzfunktion ein "Mix" auf Sitefilter und Contentfilter.

Bei dem *Sitefilter* werden bestimmte URL oder Domains in einer Datenbank festgelegt und somit verboten, der *Contentfilter* durchsucht den Inhalt (Content) einer Webseite auf verbotene Stichwörter, die der Administrator zuvor in einer Datenbank festgelegt hat.

Eine weitere sinnvolle und wichtige Funktion ist der Werbeblocker, da er einen schnelleren Aufbau von Webseiten ermöglicht. Bei den meisten Desktop Firewalls geschieht dies ebenfalls mit einer schon bestehenden Datenbank, worin schon bekannte Arten von Werbung gespeichert sind. Allerdings kann auch hier der Benutzter selbst Hand anlegen und unerwünschte Werbung wie PopUps, für den nächsten Besuch selbst definieren.

4. Schlusswort

Eigentlich ist nur zu merken, dass eine gesamte Desktop-Firewall auf einer riesen Liste von Regeln basiert, die es gilt mit viel Mühe sorgfältig zu erarbeiten.
Ich hoffe, ich konnte Ihnen die Technik der Desktop Firewall etwas näher bringen und Ihnen die grundlegenden Informationen vermitteln. Bei weiteren Fragen helfe ich Ihnen gerne weiter! Melden Sie sich einfach im Forum auf www.computer-support.org oder per E-Mail.

-- Simon 'McSchlumpf' K.

Aktuelle Viren und Würmer

- **Einleitung**
  Dieser Artikel handelt von den aktuellen Viren und Würmern Korgo, Netsky, Sasser und Sober. Es soll dem Leser etwas mehr Wissen über die Arbeitsweise der unbeliebten Lebensgenossen vermitteln, damit sie die heutigen Risiken durch solche besser verstehen können. Sollte Interesse seitens der Leser bestehen, werden in den nächsten Ausgaben weitere Viren folgen. Wir wünschen Ihnen viel Spaß beim Lesen.
- **W32.Korgo.A**
  Korgo gibt es in mehreren Varianten. Da diese sich aber kaum unterscheiden, werde ich lediglich die A-Variante erläutern.

  Auch dieser Wurm bedient sich der seit 13. April 2004 bekannten LSASS-Schwachstelle LSASS-Schwachstelle [1] in den WINNT-Systemen. (LSASS - Local Security Authority Subsystem) [1] in den WINNT-Systemen. Nach seiner Ausführung, kopiert er sich in den Windows-Systemordner. Dort ist er dann unter den verschiedensten Dateinamen zu finden, allerdings mit einer konstanten Größe von 10.752 Bytes. Nachdem er diesen Reproduziervorgang abgeschlossen hat, erstellt er einen Registry-Eintrag:

  HKEY LOCAL MACHINE\Software\Microsoft\Windows\CurrentVersion\Run\ WinUpdate = [Windows-Systemordner]\[zufälliger Dateiname].exe

  Um sich zu verbreiten, versucht er sich mit zufälligen IP-Adressen über den Port 445 zu verbinden. Die Ergebnisse dieser versuchten Verbindungen verbreitet er in IRC-Channels auf den IRC-Servern:

* gaspode.zanet.org.za
* lia.zanet.net
* irc.tsk.ru
* london.uk.eu.undernet.org
* washington.dc.us.undernet.org
* los-angeles.ca.us.undernet.org
* brussels.be.eu.undernet.org
* caen.fr.eu.undernet.org
* flanders.be.eu.undernet.org
* graz.at.eu.undernet.org
* moscow-advocat.ru
* gaz-prom.ru

W32.Korgo.A beinhaltet auch eine Backdoor-Funktion, mit der es anderen Internetbenutzern möglich gemacht wird, den Rechner zu durchforsten und Daten herunter- und hinaufzuladen. Diese Backdoor öffnet die Ports 113, 3067 und andere, die zufällig ausgewählt werden. Symantec [2] stellt ein Tool zur Entfernung des Wurms [3] zum Download bereit.

[1]http://www.microsoft.com/technet/security/bulletin/ms04-011.mspx
[2]http://www.symantec.com
[3]http://securityresponse.symantec.com/avcenter/FixKorgo.exe

- **W32.Netsky.A** Dieser Abschnitt handelt von Netsky.A, weil damit all das Übel begonnen hat. Die Angriffsziele sind nahezu alle Windows-Systeme (von Windows 9x - Windows XP). Der Wurm verbreitet sich über den Mailverkehr und die Windows-Netzwerkfreigaben. Er durchsucht alle Speicherkomponenten nach E-Mail Adressen für seine weitere Verbreitung und hält dabei Ausschau nach Dateien mit folgenden Endungen:

  MSG, OFT, SHT, DBX, TBB, ADB, DOC, WAB, ASP, UIN, RTF, VBS, HTML, HTM, PL, PHP, TXT, EML

  Der Wurm nistet sich außerdem in allen erreichbaren Medien ein, auf denen er Schreibrechte hat. Er benutzt dabei die Dateinamen

  angels.pif
  coolscreensaver.scr
  dictionary.doc.exe
  dolly_buster.jpg.pif
  doom2.doc.pif
  e.book.doc.exe
  e-book.archive.doc.exe
  eminem-lickmypussy.mp3.pif
  hardcoreporn.jpg.exe
  howtohack.doc.exe
  matrix.scr
  maxpayne2.crack.exe
  nero.7.exe
  office_crack.exe
  photoshop9crack.exe
  porno.scr
  programmingbasics.doc.exe
  rfccompilation.doc.exe
  serial.txt.exe
  sexsexsexsex.doc.exe
  strippoker.exe
  virii.scr
  winlonghorn.doc.exe
  winxp_crack.exe

  Die doppelten Endungen sind besonders für Benutzer gedacht, die Dateiendungen normalerweise nicht sehen. Sie sollen den Schein aufkommen lassen, es wäre ein Bild oder ein Word-Dokument.

  Bei der Weiterverbreitung benutzt Netsky folgende E-Mail Adressen:

  auctions@yahoo.com
  responder@ebay.com
  responder@amazon.com
  auctions@msn.com
  responder@qxl.com

  Die angehängten Dateien fangen jeweils mit "prod_info" an und anschließend folgt eine generierte Zeichenfolge. Sollte die Datei geöffnet werden, gibt der Wurm die Fehlermeldung "The file could not be opened" aus und schreibt einen Eintrag in die Registry. Dieser ist unter *HKLM\Software\Microsoft\Windows\CurrentVersion\Run* zu finden. Die Datei, die beim Systemstart geladen wird, hat den Namen "services.exe" und soll somit einen Systemservice darstellen.

Ein Removal Tool und die dazu passende Anleitung finden Sie auf der Website von Symantec.

[1]
http://securityresponse.symantec.com/avcenter/venc/data/w32.netsky@mm.removal.tool.html

- **W32.Sasser.A**
  Sasser ist ein Wurm, der über eine Sicherheitslücke, wie im Sicherheitsbulletin MS04-011 [1] beschrieben, eindringt und sich vermehrt.
  Er verbreitet sich weiter, indem er über zufällig ausgewählte IP-Adressen seine Schadensroutinen ausführt und somit nach verwundbaren Systemen sucht.

  Sasser kann auch auf den Windows-Systemen 9x und Me aktiv werden, allerdings muss der Benutzer das Programm dazu manuell starten, da die Sicherheitslücke in diesen Systemen noch nicht vorhanden war. Die befallenen Systeme sind Windows 2000 und XP.
  Dadurch, dass Sasser ziemlich viele Systemressourcen verwendet und deswegen einige Programme nicht mehr korrekt laufen können, bemerkt der Benutzer häufig einen Befall.

  Der Wurm geht nach einem einfachen Muster vor: Um zu vermeiden, dass mehrere "Sasser"-Vertreter auf dem gleichen System laufen, wird zuerst versucht eine mutex mit dem Namen "Jobaka3l" zu erstellen. Falls diese Datei bereits vorhanden ist, beendet er sich. Er hält sich selbst in in *C:\Winnt\avserve.exe* auf und fügt folgenden Eintrag in die Registry(*HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run*) ein: *"avserve.exe"="%Windir%\avserve.exe"*
  Durch die Benutzung von AbortSystemShutdown API versucht der Schädling Ausschalten und Reboots zu verhindern. Nach Aussen hin startet er zusätzlich noch einen FTP-Server hinter Port 5554, um weiter Rechner zu infizieren. Zur Verbreitung bedient er sich dem Systemdienst gethostbyname von Windows und sucht daraus IP-Adressbereiche ab. Der Wurm ignoriert dabei alle internen Netzwerke, die mit 127.0.0.1, 10.*.*.*, 172.*.*.*, 192.*.*.*, 169.*.*.* beginnen. Er generiert andere IP-Adressen, indem er die letzten beiden Elemente einer IP-Adresse durch zufällige Zahlen zwischen 1 und 255 ersetzt oder er verwendet komplett zufällige Adressen. Danach untersucht er den Port 445 auf dem neuen Zielrechner, um zu testen, ob dieser online ist.

  Anschließend exploitet Sasser durch einen Stack-basierten Bufferoverflow den Lsass-Dienst und öffnet auf ungepatchten Systemen eine Shell auf Port 9996, worüber er Shellcodes an den Angreiferhost sendet. Diese Shellcodes veranlassen, dass sich die Shell nun zurück auf den FTP-Server des Angreiferhosts verbindet und sich alle benötigten Dateien holt, um den Wurm zu installieren. Dies geschieht durch eine exe-Datei, die normalerweise 5 frei gewählten Zahlen beginnt und dann mit _up.exe endet. Der Wurm erstellt noch ein File C:\win.log, in dem er protokolliert, über welche IP er hineingekommen ist und wie viele Rechner er bereits infiziert hat. Nach dem Beenden der Shell, stürzt der Lsass.exe-Prozess über einen korrupten exit-Befehl ab und löst eine Fehlermeldung aus, bevor der Rechner innerhalb der nächsten 60 Sekunden heruntergefahren wird.

  Um diesen lästigen Parasiten loszuwerden sind folgende Schritte notwendig:
  Zurest sollte man den PC von allen Netzwerken trennen, damit ein Neubefall ausgeschlossen ist. Zur Not kann man auch das Kabel ausstecken. Danach den Computer neustarten und wenn Windows komplett geladen ist, die 'MS-Dos Eingabeaufforderung' öffnen, indem man auf Start -> Ausführen geht und dort *cmd* eingibt und mit Return bestätigt. Dort gibt man nun '*shutdown -i*' ein und drückt wieder Enter. Jetzt sollte der sog. 'Remote Shutdown Dialog' kommen, in welchem man zurest auf "Hinzufügen" geht und den Computernamen in das hierfür vorgesehen Feld einträgt. Danach mit [OK] bestätigen. In dem Warnfeld nun 9999 eingeben und in das Kommentarfeld: *Delay Lsass.exe shutdown* eingeben und daraufhin wieder auf [OK].
  Jetzt wieder mit dem Netzwerk verbinden und den Patch[0] von Microsoft installieren. Danach sollte man noch das 'Removal Tool'[2] von Symantec verwenden um den Wurm entgültig auszurotten.

[1]http://www.microsoft.com/technet/security/bulletin/MS04-011.mspx
[2]http://securityresponse.symantec.com/avcenter/FxSasser.exe

- **W32.Sober.C**
  Sober ist ein E-Mail-Wurm, der seinen eigenen SMTP-Server mit sich bringt und sich somit ohne Probleme weiterverbreiten kann. Er verändert seine Betreffzeile von Mail zu Mail. Die Betreffszeilen nimmt er aus einer integrierten Liste, die aus englischen, aber auch deutschen Sätzen besteht.
  Der Anhang verändert auch von Mail zu Mail seinen Namen, allerdings bleibt es eine *.bat, .com, .cmd, .exe, .pif* oder *.scr* Datei.

  Sobald der Wurm ausgeführt wird, zeigt er eine komische Fehlermeldung an. Man denkt er wäre beendet worden.

  Der Wurm ist in Visual Basic programmiert, mit UPX komprimiert, und erstellt von sich drei Kopien im Windows-Systemordner. Eine von den Dreien heisst immer *similare.exe* und die beiden weiteren werden zufällig generiert.

  Um sich selbst beim Systemstart auszuführen, fügt er in der Registry (*HKLM\Software\Microsoft\Windows\CurrentVersion\Run*) einen Eintrag mit seinem Dateinamen hinzu.

  Die mutierte Version W32.Sober.A erstellt im Windows-Systemordner noch eine Datei *Macromed\Help\Media.dll*, in der alle auf dem befallenen System gesammelten E-Mail-Adressen eingetragen sind. Die Datei kann ohne Probleme gelöscht werden, da sie unwichtig ist. Vielleicht kann man sich auch noch bei den Empfängern der Virus entschuldigen. ;)

  Er verwendet eine spezielle Technik um sich neuzustarten, sollte der Prozess beendet werden.

  Mögliche Dateinamen sind:
  anti-Sob.bat Anti-Sob.bat
  anti-trojan.exe
  anti_virusdoc.pif
  AntiTrojan.exe
  AntiVirusDoc.pif
  Bild.scr
  check-patch.bat
  Check-Patch.bat
  CM-recover.com
  CM-Recover.com
  funny.scr
  Funny.scr
  Hengst.pif
  Liebe.com
  little-scr.scr
  love.com
  Mausi.scr
  nacked.com
  NackiDei.com
  Odin_Worm.exe
  perversion.scr
  Perversionen.scr
  pic.scr
  playme.exe
  potency.pif
  Privat.exe
  private.exe

removal-tool.exe
Removal-Tool.exe
robot_mail.scr
robot_mailer.pif
RobotMailer.com
schnitzel.exe
screen_doc.scr
Screen_Doku.scr
security.pif

Mögliche Betreffe sind:
New internet virus!
You send spam mails (Worm?)
A worm is on your computer!
Now, its enough
You have sent me a virus!
Hi darling, what are you doing now?
Be careful! New mail worm
Re: Contact
RE: Sex
Sorry, Ive become your mail
Hey man, long not see you
Re: lol
Viurs blocked every PC (Take care!)
Surprise
Ive become your mail!
Advise who I am!
New Sobig-Worm variation (please read)
Back At The Funny Farm
I love you (Im not a virus!)
Neuer Virus im Umlauf!
Sie versenden Spam Mails (Virus?)
Ein Wurm ist auf Ihrem Computer!
Langsam reicht es mir
Sie haben mir einen Wurm geschickt!
Hi Schnuckel was machst du so ?
VORSICHT!!! Neuer Mail Wurm
Re: Kontakt
RE: Sex
Sorry, Ich habe Ihre Mail bekommen
Hi Olle, lange niks mehr geh
Re: lol
Viurs blockiert jeden PC (Vorsicht!)
_berraschung
Ich habe Ihre E-Mail bekommen !
Jetzt rate mal, wer ich bin !?
Neue Sobig Variante (Lesen!!)
Back At The Funny Farm
Ich Liebe Dich

Und der in der E-Mail enthaltene Text kann einer der Folgenden sein:

"Congratulations!! Your Sobig Worms are very good!!!
You are a very good programmer!
Yours faithfully
Odin alias Anon"

"Kaspersky Lab Int. and Norton Anti Virus have found a new typ of worm.
He calls itself "ODIN" and he is very variable!
The worm hides in the screen saver.

Read the -screen_doc- documentation and you will be able to
find and kill this virus!",

"I permanently get Spam-Mails from you and inside is a virus!!
You should remove these thing.
Sorry, but the ODIN Worm is probably on your computer!
You should check this with the patch application.

See you soon",

"Automatic Mail notification: Robot-System__##

Answer = complete %Error% occured%

Answer transferred in attachement -Access",*

"Or are you put under stress?
I,, I put only under stress,,, every sec, min, hour, day,.....
You see, I've an another mail-name!
But, it's too dangerous to say it,, here in the internet.
Every can read my problems! Use the attach.,
the password is your birthday.

See you soon!",

"Sorry :-) it's late,, I know,, but I`ve a new mail adress.
I've got my own screen saver;; with me!
Other say, it`s nice, but,,... see self.
Ok ok ,, I'm nacked in this pic, but, it is a work of art!
Yaya I know i know!",

"I hope you know of me!
When not, please delete this mail!",

"New Sobig variation in the net.
Save yourself with the patch before it's too late!
The new Sobig is very dangerous!",

"Actually, this bastardos have installed a trojan on my computer!
And now, I'm here,.,. I've tell you something about the..
No, not here, I'll to report you,, next days!
But before, you must check your system. Trojan are everywhere!!!
Check first your system with the tool.
see ya",

"You must change any settings before the worm control your computer!
But, read the official statement from Norton Anti Virus!",

"Sorry, but the ODIN Worm is probably on your computer!
You should check this with the patch application.
See you soon",

"Kaspersky Lab Int. and Norton Anti Virus have found a new typ of worm.
He calls itself and he is very variable!
This mail was spread with this Worm, too. BUT, the attachement is a AntiVirus!!",

"Automatic Mail notification: Robot-System__##
WHEN YOU CAN NOT READ THIS MAIL ATTACH.,
PLEASE REPORT US THIS ERROR.",

Die angegebenen Virenwarnungen vor der Wurmfamilie W32.Sobig sind sogenannte Hoaxes und bedürfen keiner weiteren Beachtung. Um den Worm vom heimischen Rechner zu bannen, muss man sie das "Sober Removal Tool" von Symantec [1] runterladen, und anschließend sollte man sich die Anleitung zur Entfernung [2] ausdrucken. Wenn man die Anleitung befolgt, sollte es keine Probleme mehr geben.

[1]http://securityresponse.symantec.com/avcenter/FixSober.exe
[2]http://securityresponse.symantec.com/avcenter/venc/data/w32.sober.removal.tool.html

-- Peter Wilfahrt und Gerrit 'Notro' E.

Wireless Security Authentications

"WEP and other Authentication Methods"
Sources: airsnort-2.1b.tar.gz

1 WEP

Wired Equivalent Privacy; wie der Name schon sagt, soll WEP einen vergleichbaren Schutz gegen unerlaubtes Abhören des Wireless LAN's bieten, wie dies bei einem kabelgebundenen Netzwerk der Fall ist. Deshalb bietet der 802.11 Standard mit WEP die Möglichkeit, Daten zu verschlüsseln. Leider hat WEP einen Designfehler, weshalb es einem Attacker möglich ist, die Verschlüsselung zu brechen und sich so Zugang zum Netz zu verschaffen.

1.1 RC4

WEP basiert auf dem Verschlüsselungsalgorithmus RC4 von Ron Rivest der RSA Data Security, Inc. Die genaue Funktion von RC4 wird später noch genau erklärt.

1.2 IV-Reuse

Weil RC4 als Stream-Cipher anfällig auf Synchronisationsfehler ist und bei einer Funkübertragung eine hohe Fehlerwahrscheinlichkeit herrscht, wurde entschieden, dass der Stream für jedes Paket neu initialisiert wird. Dieser „Missbrauch" von RC4 führt zu einem fatalen Problem. Das grundlegende Problem liegt darin, dass bei einer Stream-Cipher wie RC4 niemals derselbe Strom zweimal verwendet werden darf. Diese kann später verwendet werden, um bei einem erneuten Auftauchen des IV sofort das Paket zu entschlüsseln. Dieser Angriff ist zwar praktikabel, jedoch sehr aufwändig. Mit diesem Angriff ist es allerdings nicht möglich, den WEP-Key zu ermitteln.

1.3 FMS-Attacken

Diese Attacke ist viel wirkungsvoller, aber auch erheblich komplizierter zu verstehen; sie wurde von Scott Fluhrer, Itsik Mantin und Adi Shamir entdeckt. Sie nutzt die Schwäche im Key Scheduling Alghoritmus aus. Die Attacke basiert darauf, dass das erste word des Key-Streams überproportional vom Schlüssel abhängt. Es ist daher möglich, eine statistische Aussage über den Schlüssel zu machen. Wenn wir genügend Pakete sammeln, können wir die möglichen Schlüssel immer mehr einengen und letztendlich werden wir den Schlüssel erhalten. Das Ganze ist sehr mathematisch und ich verzichte deshalb darauf, es hier im Detail zu erläutern.

WEPCrack und AirSnort basieren auf dieser Attacke. Diese können in verhältnismäßig kurzer Zeit (5-10 Mio Pakete) den Schlüssel ermitteln.

1.4 WPA & 802.11i (WEP Weiterentwicklungen)

Wi-Fi Protected Access soll die Schwächen von WEP beheben. Es wurde von der WiFi-Alliance entwickelt, einer Gruppe von verschiedenen Herstellern. Es basiert auf den Entwürfen zum IEEE Standard 802.11i. Die Idee ist schnellst möglich eine bessere Alternative zu WEP zu bieten und dabei den zukünftigen Standard vorzubereiten. WPA soll mit Firmware Updates auf bestehender Hardware funktionieren.

WPA kann in zwei Hauptelemente aufgeteilt werden: zum einen TKIP, welches die Verschlüsselung verbessern soll, und zum anderen 802.1X zur verbesserten Authentisierung.

1.4.1 TKIP

TKIP steht für Temporal Key Integrity Protocol. TKIP ist ein Wrapper, der WEP umschließt und so ausgelegt ist, dass dies mit bestehender Hardware realisiert werden kann. Es ist als eine Art Bugfix, der Schwächen von WEP beheben soll, zu verstehen:

1. *Michael*, ein kryptografischer Message-Authentication-Code, zur Verhinderung von

gefälschten Paketen.
2. *IV Sequenz*, um replay-Attacken zu verhindern.
3. Ein Schlüssel *Mix-Funktion*, um einen neuen Schlüssel für jedes Paket zu generieren.
4. *Re-Keying* Mechanismus, um regelmäßig neue Schlüssel zu generieren.

1.4.2 802.1X (EAP)

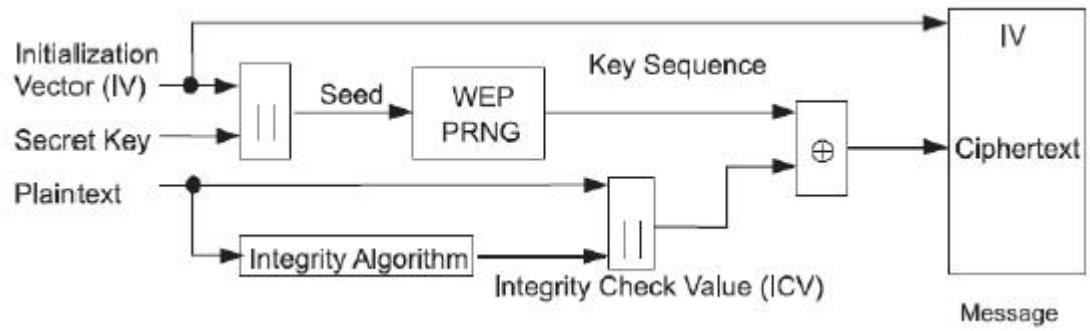
Der IEEE 802.1X Standard dient zur Authentisierung und zum Schlüsselmanagement. Der Standard wurde ursprünglich für kabelgebundene IEEE 802 LANs entwickelt; als solches ist nicht festgeschrieben in welcher Form es für Drahtlose 802.11 Netzwerke eingesetzt werden kann. 802.11X basiert auf EAP (Extensible Authentication Protocol) und definiert als solches keine feste Methode zur Authentisierung. EAP wurde für PPP Verbindungen entwickelt und bietet die Möglichkeit, nachdem der Link steht, die Authentisierung vorzunehmen. Als solches bildet es ein Framework, auf dem eine beliebige Authentisierung implementiert werden kann. Die gängigste Methode ist die Implementierung mit RADIUS. Dabei bietet es eine sogenannte Port-Based Authentication, was soviel heißt wie, dass der Port, in WLAN-Fall eine Assoziation mit dem Access Point, authentisiert wird. Ist dies geschehen, hat dieser Port vollen Zugang zum Netzwerk. (Port ist hier nicht mit einem TCP/IP-Port zu verwechseln).

LEAP steht für Lightweight EAP und ist ein Cisco Implementation von EAP mit RADIUS als Authentication Meachanismus, welche in den Cisco Aironet Access Point implementiert ist.

1.5 WEP

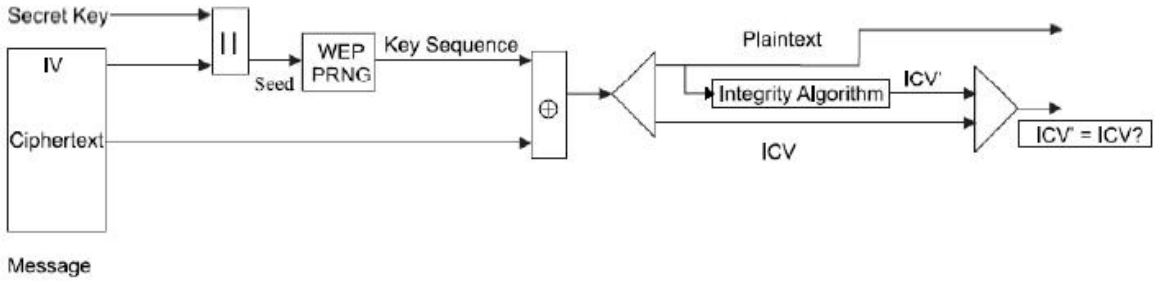
WEP ist eine Form von Electronic Code Book Algorithmus, dabei wird der Plaintext mittels XOR mit einer pseudo-zufälligen Schlüsselsequenz der gleichen Länge kombiniert.

Die folgende Grafik zeigt den kompletten Verschlüsselungsvorgang. Zum einen wird der *secret key*, der über einen geheimen Kanal erst ausgetauscht werden muss, mit einem IV (Initialisierungs-Vektor) kombiniert und dient als *seed* für den WEP-PRNG (Pseudo Random Number Generator)



Auf der anderen Seite wird über Plaintext der sogenannte *integrity check value* berechnet, um die Integrität der Daten zu garantieren. Dieser ICV wird dann auf dem Plaintext abgehängt und mit dem Schlüsselstrom mittels XOR kombiniert. Das Resultat des XOR Vorgangs ist der *ciphertext*, welcher zusammen mit dem verwendeten IV die *message* ergibt, welche übertragen wird.

Als PRNG verwendet WEP den RC4 Algorithmus. Der ICV ist eine einfache CRC-32 Checksumme. Folgende Grafik zeigt die Entschlüsselung einer ankommenden Nachricht:



Mit Hilfe der *secret keys* unter dem IV aus der *message* wird wieder die Schlüsselsequenz erzeugt. Diese wird mit dem *ciphertext* per XOR kombiniert. Zum Schluss wird noch der ICV über dem entschlüsselten Text errechnet und mit dem gelieferten Wert verglichen.

1.6 RC4

RC4 ist eine Strom-Chiffrierung mit variabler Schlüssellänge. Es ist der am weitesten verbreiteten Algorithmus und wird beispielsweise auch bei SSL eingesetzt.

Der Aufbau von RC4 ist sehr einfach und deshalb auch leicht zu implementieren, zudem ist er sehr schnell. Es hat eine 8x8 S-Box S_0, S_1, S_2…$S_{255.}$ Die Einträge sind Permutationen von 0-255 und die Permutation wird vom Schlüssel erzeugt.

Die S-Box wird folgendermaßen initialisiert: Zuerst wird die S-Box linear gefüllt: $S_0 = 0$ und $S_1 = 1$ etc., dann wird ein zweites 256-Byte Array mit dem Schlüssel gefüllt, wozu der Schlüssel falls nötig wiederholt wird (K_0, K_1…K_{255}).

$j = 0$
for $i = 0$ **to** 255
  $j = (j + S_i + K_i) \bmod 256$
  swap S_i and S_j

Der Strom wird nun folgendermaßen erzeugt:

$i = 0$
$j = 0$

$i = (i + 1) \bmod 256$
$j = (j + S_i) \bmod 256$
swap S_i and S_j
$t = (S_i \text{ and } S_j) \bmod 256$
$K = S_t$

K ist nun das nächste Byte im Schlüsselstrom, der Vorgang wird für jedes weitere benötigte Byte wiederholt.

1.7 Erklärung zu TKIP

Im Folgenden werde ich euch noch die einzelnen Element von TKIP erklären.

Michael Michael ist ein völlig neuentwickelter Message-Authentication-Code, wobei zwei Punkte wichtig beim Design waren: zum einen musste er eine genügend große Sicherheit bieten und zum anderen mit sehr wenig Rechenleistung auskommen, da er auch auf Hardware funktionieren sollte, die dazu nicht dimensioniert wurde.

Der Michael Schlüssel ist 64-bit lang, dargestellt als 2x 32-Bit little-endian words (K_0, K_1). Die Michael tagging Funktion füllt als erstes die Nachricht mit hex 0x5a und genügend null Füll-Bytes das die Gesamtlänge ein vielfaches von 32 Bytes ergibt. Anschließend wird das Resultat in 32-bit Wörter zerlegt M1, M2, M3… Mn. Zum Schluss wird das Tag mit Hilfe des Schlüssels

und der Message-Wörter nach folgender Struktur errechnet:

(L, R) ← (K0, K1)
do *i* **from** 1 **to** *n*
L ← L ⊕ M_i
(L,R) ← *b*(L, R)
return (L, R) as the tag

⊕ stellt ein exclusive or (XOR) dar und *b* ist eine einfach Funktion basierend auf Rotationen, Addition und Bit Tausch.

IV Sequenz Mit Hilfe von Michael kann zwar verhindert werden, dass Pakete als Ganzes gefälscht werden, aber er kann keine Replay Attacke verhindern. Um dies zu verhindern, verwendet TKIP den Initialisierungs-Vektor (IV) als Sequenzzähler. Dazu wird der IV bei jedem Schlüsselwechsel neu initialisiert und dann mit jedem Paket erhöht.

1.8 References

[1] Bernard Aboda. The unofficial 802.11 security web page. http://www.drizzle.com/~aboba/IEEE/

[2] Nikita Borisiv, Ian Goldberg and David Wagner. Intercepting mobile communications: The insecurity of 802.11. In *7th Annual International Conference on Mobile Computing* and Networking, 2001. http://www.isaac.cs.berkeley.edu/isaac/mobicom.pdf

[3] EAP Working Group. Extensible authentication protocol (eap). RFC 2284. http://www.ietf.org/internet-drafts/

[4] Cameron MacNally. Cisco leap protocol description. http://www.missl.cs.umd.edu/wireless/ethereal/leap.txt

[5] Daniel Walther. How to setup Airsnort under Windows. http://www.wireless-bern.ch/modules.php?op=modload&name=Reviews&file=index&req=showcontent&id=8

[6] LAN/MAN Standards Committee of the IEEE Computer Siciety. Port-based network access control. IEEE 802.1X, 2001. http://standards.ieee.org/getieee802/download/802.1X-2001.pdf

[7] Bruce Schneider. *Applies Cryptography*. John Wiley & Sons, second edition, 1996.

[8] Adam Subblefiled, John Ioannidis and Aviel D. Rubin. Using the fluhrer, mantin, and shamir attack to break wep. Technical report, AT&T Labs Research, Auhust 2001. http://www.cs.rice.edu/

[9] Jesse Walker. 802.11 security series. 2002. http://www.intel.com/cd/ids/developer/asmo-na/eng/index.htm

[10] WiFi-Alliance. Wi-Fi protected access (wpa) security web page. http://www.wifialliance.com/OpenSection/protected_access.asp

-- Simon 'Zodiac' Moser

Überblick über Portscan-Techniken

Inhalt

Vorwort

Dieser Text soll einen Überblick über simple Portscan-Techniken geben. Wer nun denkt, das habe ich schonmal irgendwo gelesen, liegt wohl richtig. Ich habe diesen Text ursprünglich nicht als Artikel für das *Spaxid* geschrieben, eher als Zusammenfassung für mich selbst. Ich habe mich dabei sehr an Randy Williams[1] Artikel Low Level Enumeration[2] orientiert der auf SecurityFocus.com erschienen ist. Wer Interesse hat und des Englischen mächtig ist, sollte sich diesen Text ebenfalls einfach mal anschauen. Ich habe daraus das für mich wichtigste zusammengefasst.

Einleitung

Mittlerweile gibt es immer mehr und komplexere Techniken Ports zu scannen. Wer sich mit der Thematik wenigstens etwas mehr auseinandergesetzt hat, als mal schnell zu schauen, ob die Firewall funktioniert, wird wohl um Nmap[3] nicht herum gekommen sein. Um damit ernsthaft zu arbeiten, vernünftig eine Firewall zu testen, sollte man wissen, wie die einzelnen Scans arbeiten und was dahinter steckt. Dieses Paper bietet eine kleine Einführung und einen meiner Meinung nach vernünftigen Überblick darüber.

Syn-Scans

Der Syn-Scan wurde vor ein paar Jahren entwickelt und ist die wohl mittlerweile am verbreitesten Scan-Technik, obwohl es „stealthere" Methoden gibt. Dieser Scan, auch als "halb offener Scan" (oder „half open") bekannt, arbeitet damit, Verbindungen noch während der Anfrage abzubrechen. Das Betriebssystem verwirft die Verbindung einfach. Es wird darauf gesetzt, dass dieser Scan in der Regel nicht von Firewalls erkannt, nicht geloggt und einfach als Verbindungsfehler verworfen wird. Er ist die wohl sinnvollste Kombination aus Unsichtbarkeit und Geschwindigkeit.

Um zu verstehen, wie diese Lücke von Scannern ausgenutzt wird, sollten wir kurz einfache TCP/IP Grundlagen klären. TCP Pakete erhalten ihre Aufgabe über sogenannte Flags im Paketheader:

SYN – Synchronisieren, Verbindungsaufbau forden
ACK – Bestätigung (Acknowledgement), Verbindungsaufbau bestätigen
FIN – Ende (Finish), teilt dem Host/Server mit das keine Pakete mehr folgen
RST – Reset, um eine Verbundung zu reseten

Um nun den SYN Scan zu verstehen, muss man wissen, wie ein TCP/IP Verbindungsaufbau funktioniert. Dieser geht nach dem Three-Way-Handshake vonstatten.

Dieses kleine ASCII Diagramm verdeutlicht den Vorgang wohl am besten:

SYN

```
Client ----------------> Server
          SYN/ACK
Client <---------------- Server
            ACK
Client ----------------> Server
```

Der Client schickt ein SYN-Paket zum Server und signalisiert damit, dass er eine Verbindung aufbauen will. Der Server antwortet mit einem RST, wenn er die Verbindung nicht annehmen will, oder mit einem SYN/ACK, wenn er bereit für den Verbindungsaufbau ist. Erhält der Client das SYN/ACK-Paket, antwortet er mit einem ACK, um dem Server mitzuteilen, dass die Verbindung steht. Nun können Daten ausgetauscht werden.

Der SYN Scan macht sich dieses Verfahren zu Nutzen. Er schickt ein SYN-Packet an den Server, dieser gibt, wenn der Port offen ist, ein SYN/ACK zurück; ist er geschlossen, ein RST.
Um jedoch den Scan unsichtbar zu halten, wird der 3. Teil des Handshakes einfach nicht durchgeführt, sondern mit einem RST abgebrochen. Der Server denkt, es wäre ein Verbindungsfehler und verwirft im Normalfall einfach die Verbindungsaufforderung.

Auch SYN DoS-Attacken funktionieren nach diesem Verfahren, da sie den Server mit halboffenen Verbindungen flooden, also quasi zumüllen, und somit daran hindern, normale Verbindungen aufzubauen.

Beim Portscannen mit der SYN Technik sollte man einen vernünftigen Rythmus wählen und nicht blind alle Ports scannen, da sonst ein ungewollter DoS begünstigt wird. Das Ziel ist es, unsichtbar zu bleiben, was man mit Sicherheit nicht erreicht, wenn der gescannte Host keine Verbindungen mehr annimmt.

Xmas, FIN und Null Scan
Xmas und FIN Scanns sind die wohl von IDS (Intrution Detection Systems) am wenigesten erkannten Scanarten, leider aber auch die unzuverlässigsten. Der Grund dafür ist, dass sie unter vielen Faktoren den Scanner dazu bringen können, einen Port als offen zu sehen, wenn er es definitiv nicht ist.

Xmas
Hier wird einfach ein TCP-Paket an den Server geschickt, bei dem die FIN/URG/PSH Flags gesetzt sind. Wenn der Port offen ist, wird der Server nichts zurückgeben; ist der Port geschlossen, sendet der Server ein Packet zurück bei dem die RST/ACK Flags gesetzt sind. Dies gillt bei allen Betriebssystemen deren TCP Implementierung auf dem Standard (RFC 793[4]) für TCP basiert.
Da Microsoft sich wie so oft nicht an einen Standard hält und seinen eigenen Kram macht, bekommt man bei allen Windows Betriebsystemen nur geschlossene Ports als Ergebnis zurück. Windows ist daher nicht für einen Xmas Scan anfällig.

FIN und NULL Scan
Beide Scans arbeiten eigentlich genauso wie der Xmas Scan. Der FIN Scan hat in den Paketen ein FIN Flag gesetzt, der Null Scan gar kein Flag. Ist der Port geöffnet, kommt nichts zurück; ist er geschlossen, ein Paket bei dem die Flags RST/ACK gesetzt sind. Wie beim Xmas Scan sind nur Betriebssysteme für diese beiden Techniken anfällig, die dem RFC Standard entsprechen. Somit lässt sich jedoch wenigstens sagen, was das Betriebsystem des Hosts ist, den man scannt, zumindest lässt es sich grob eingrenzen.
Wenn sich jemand für OS-Detection interessiert, sollte er einen Blick auf den Artikel[5] von Fyodor[6] werfen, der im Phrack54 erschienen ist. Eine deutsche Übersetzung[7] gibt es auf Insecure.org.

Ein weiteres Problem bei Xmas, FIN und Null Scans sind Firewalls. Normalerweise geben diese kein RST zurück, sondern droppen(=verwerfen) die Pakete einfach. Somit sieht es dann aus, als seien alle Ports geöffnet. Trotzdem sind diese Techniken eine einfache, mächtige und unsichtbare Methode Systeme zu scannen.

Idle oder "Zombie" Scans

Kommen wir nun zu einer interessanten Scantechnik, die auf eine etwas andere Art und Weise arbeitet als die vorherigen. Diese Technik ist relativ neu und wahrscheinlich auch die "unsichtbarste". Kein Paket mit der IP des Rechners, der scant, gelangt überhaupt zum Server. Sie nutzt den Umstand aus, wie IP-IDs vom Betriebsystem vergeben werden. Jedes IP-Paket, das von einem Host gesendet wird, bekommt normalerweise eine ID. Die meisten Betriebsysteme inkrementieren die Paket-ID nach einer Verbindung ganz einfach. Der Angreifer kann so eine überraschend genauen Überblick bekommen, was auf dem Server vorgeht.

Die ID-Vergabe prüfen wir ganz einfach mit einigen Pings auf den Server. Anhand der ID der Antwortpakete sehen wir, wie diese vergeben wurde.
Die Pakete können mit jedem beliebigen Sniffer wie zum Beispiel Ethereal[8] ausgewertet werden. In einem Beispiel benutzen wir hping[9], ein sehr flexibles Tool für die Kommandozeile mit dem man schnell und einfach unterschiedliche Pakete generieren kann:

```
root@john-doe echox # hping2 -1 nobody
HPING nobody (eth0 192.168.2.2): icmp mode set, 28 headers + 0 data bytes
len=46 ip=192.168.2.2 ttl=64 id=54481 icmp_seq=0 rtt=0.3 ms
len=46 ip=192.168.2.2 ttl=64 id=54482 icmp_seq=1 rtt=0.2 ms
len=46 ip=192.168.2.2 ttl=64 id=54483 icmp_seq=2 rtt=0.2 ms
len=46 ip=192.168.2.2 ttl=64 id=54484 icmp_seq=3 rtt=0.2 ms
--- nobody hping statistic ---
4 packets tramitted, 4 packets received, 0% packet loss
round-trip min/avg/max = 0.2/0.2/0.3 ms
```

Wie hier zu sehen ist, wird die ID einfach fortlaufend vergeben. Was passiert nun, wenn der Server in der Zwischenzeit andere Verbindungen hat? Prüfen wir das einfach nochmals durch ein paar weitere ICMP-Pakete:

```
root@john-doe echox # hping2 -1 nobody
HPING nobody (eth0 192.168.2.2): icmp mode set, 28 headers + 0 data bytes
len=46 ip=192.168.2.2 ttl=64 id=54488 icmp_seq=0 rtt=0.3 ms
len=46 ip=192.168.2.2 ttl=64 id=54489 icmp_seq=1 rtt=0.2 ms
len=46 ip=192.168.2.2 ttl=64 id=54492 icmp_seq=2 rtt=0.2 ms
len=46 ip=192.168.2.2 ttl=64 id=54494 icmp_seq=3 rtt=0.2 ms
--- nobody hping statistic ---
4 packets tramitted, 4 packets received, 0% packet loss
round-trip min/avg/max = 0.2/0.2/0.3 ms
```

Wie wir hier sehen können, wurde die ID beim 3. Paket um drei Schritte erhöht, beim 4. Paket um zwei Schritte. Was ist hier nun passiert? In der Zwischenzeit wurden ganz einfach Verbindungen aufgebaut oder Pakete zu anderen Rechnern verschickt. Wir können also feststellen, ob zwischen Paket 1 und Paket 2 noch andere Pakete angekommen oder Antworten raus sind. Für diesen Scan brauchen wir nun zusätzlich irgendeinen anderen Server, der auf ein Ping antwortet und am besten normalerweise keinen Traffic bekommt. Er dient uns als "Zombie". Erklären lässt sich die Technik am besten an einem Beispiel. Nehmen wir einfach an, wir wollen prüfen, ob beim Server foo.bar der Port 25 offen ist. Wir haben einen Server gefunden, der irgendwo in Alaska in der Wüste steht, nur einmal im Monat Pakete bekommt und den Namen will.net hat ;-). Ihn benutzen wir als Zombie.

Schicken wir nun ein gespooftes Paket an foo.bar mit der IP unseres Zombies. Foo.bar denkt nun, er hat ein Paket mit Destination-Port 25 von will.net erhalten.
Ist der Port von einer Firewall gefiltert und wird das Paket verworfen, schickt foo.bar nichts zurück. Die IDs von will.net werden nicht erhöht.
Ist der Port nicht gefiltert sondern einfach nur geschlossen, wird foo.bar einfach ein RST zurückschicken, die ID's von will.net werden nicht erhöht.
Ist der Port offen, wird eben mit dem Three Way Handshake geantwortet, also ein SYN/ACK zurückgeschickt. Da aber will.net von einem SYN-Paket, das er foo.bar angeblich geschickt haben soll, nichts weiß, wird er mit einem RST antworten und seine ID um eins erhöhen.

Pingen wir nun die ganze Zeit über noch will.net und beobachten die IDs der Pakete, lässt sich sehr leicht feststellen, wann unser Zombie das SYN/ACK Paket erhält und mit RST die Verbindung wieder abbricht. Das ganze würde dann so aussehen:

```
root@john-doe echox # hping -S -p 25 -a nobody deepblue
HPING 192.168.2.1 (eth0 192.168.2.1): S set, 40 headers + 0 data bytes

--- 192.168.2.1 hping statistic ---
6 packets tramitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms

root@john-doe echox # hping2 -1 nobody
HPING nobody (eth0 192.168.2.2): icmp mode set, 28 headers + 0 data bytes
len=46 ip=192.168.2.2 ttl=64 id=54488 icmp_seq=0 rtt=0.3 ms
len=46 ip=192.168.2.2 ttl=64 id=54489 icmp_seq=1 rtt=0.2 ms
len=46 ip=192.168.2.2 ttl=64 id=54490 icmp_seq=2 rtt=0.2 ms
len=46 ip=192.168.2.2 ttl=64 id=54494 icmp_seq=3 rtt=0.2 ms
--- nobody hping statistic ---
4 packets tramitted, 4 packets received, 0% packet loss
round-trip min/avg/max = 0.2/0.2/0.3 ms
```

In diesem Beispiel wäre der Zombierechner nobody, der Host einfach deepblue.

EOF

So, das wars. Ich hoffe, eine übersichtliche Einleitung in das Thema ist mir gelungen. Wem es bisher noch nicht klar ist: Dies ist nur die Spitze des Eisbergs was Portscans angeht. Wer sich näher damit beschäftigen möchte, sei als weitere Literatur die Texte auf SecurityFocus.com und Insecure.org empfohlen. Kritik, Fragen, Verbesserungsvorschläge bitte per E-Mail *(PGP-Key ist auf http://spaxid.it-helpnet.de zu finden)*.

Fußnoten

[1] Randy Williams <dolph -at- woh.rr.com>
[2] http://www.securityfocus.com/guest/24226
[3] http://www.insecure.org/nmap
[4] RFC 793 - Transmission Control Protocol: http://www.faqs.org/rfcs/rfc793.html
[5] http://www.insecure.org/nmap/nmap-fingerprinting-article.html
[6] <fyodor -at- insecure.org>
[7] http://www.insecure.org/nmap/nmap-fingerprinting-article-de.html
[8] http://www.ethereal.com
[9] http://www.hping.org

-- Simon 'echox' Koelsch

Kryptologie – Teil 1

Dieser Text dient als Grundlage für spätere Ausgaben

1. Vorwort

Verschlüsselung. Was ist das? Wozu brauche ich das? Und wie funktioniert es? Diese Fragen mag sich vielleicht ein kleiner Teil der Leserschaft stellen. Doch genau hier soll mein erster Bericht über dieses Thema beginnen. Ich möchte nämlich mit jedem weiteren Text nach und nach immer tiefer in die Thematik der Kryptologie eindringen und die Teilgebiete dieser Wissenschaft auch für Leute, die sich nicht dauernd mit dieser auseinander setzen, erklären und verständlich darstellen. Natürlich wird man auch hier nicht an der Mathematik vorbeikommen, doch ich werde versuchen, die einzelnen Schritte so verständlich wie möglich darzustellen. Der erste Bericht soll nun aber nicht direkt mit mathematischen Formel beginnen, sondern erst einmal mit den einfachsten Grundlagen beginnen und klären, was die Kryptologie überhaupt ist, wozu sie gebraucht wird und dazu noch die historischen Anfänge dieser Wissenschaft darstellen. Da ich kein absoluter Experte auf diesem Gebiet bin, kann es gut sein, dass mir auch Fehler unterlaufen, für die ich mich schon jetzt entschuldigen möchte. Wenn Ihnen ein Fehler auffallen sollte, dann bitte ich Sie darum, mir diesen mitzuteilen. Des Weiteren kann ich nicht alle Teilgebiete, die der ein oder andere unter diesem Schwerpunkt gerne lesen möchte mit in diesen Bericht einbringen, da der Text ansonsten zu lang werden würde. Somit habe ich manches rausgelassen, aber vielleicht wird dies in einem der weiteren Texte berücksichtigt werden. Danke.

2. Kryptologie - was ist das?

Die Kryptologie setzt sich aus der Kryptographie und der Kryptoanalyse zusammen. Die Kryptographie ist die Wissenschaft von der Verschlüsselung einer Mitteilung oder von der Verschleierung des Inhalts einer Mitteilung. Manchmal wird der Begriff der Kryptographie allgemeiner im Sinne von Kryptologie gebraucht und bezeichnet alles, was mit der Ver- und Entschlüsselung zu tun hat. Noch nie war die Kryptographie so bedeutend wie heute. Überall wo man hinschaut wird verschlüsselt. Dies beginnt bei einfachen geheimen Botschaften von Grundschülern, über Pay-TV-Sender, Bankautomaten bis hin zur Natur, wie z.B. bei der DNS. Überall dort werden Informationen verschlüsselt und ziemlich oft kommen sehr unterschiedliche Verschlüsselungstypen zum Einsatz.
Die Kryptoanalyse ist die Wissenschaft von der Entschlüsselung ohne Kenntnis des Schlüssels. Diese ist also das absolute Gegenteil zur Kryptographie.

Nicht nur die Menschen der letzten Jahrzehnte setzen die Kryptographie ein, sondern sie wurde schon lange vor dem Computerzeitalter angewandt, um geheime Daten zu schützen. So ist in alten Schriften zu lesen, dass schon bei Herodot im 5. Jahrhundert vor Christi erste Verschlüsselungsverfahren benutzt wurden. Der Überlieferung nach soll sie die Griechen vor der Eroberung Xerxes, dem großen König der Perser, geschützt haben. Herodot, ein im Exil lebender Grieche, benutzte dabei eine kleine Schreibtafel, von der er das Wachs abschabte, auf das Holz seine Mitteilung schrieb und dann eine neue Wachschicht über die Botschaft goß. Somit wurde sichergestellt, dass die Schrift für das bloße Auge unsichtbar blieb und der Bote überbrachte daraufhin unbehelligt die Tafel. Nach erster Verwirrung über die kuriose Lieferung kamen die Empfänger schließlich auf die Idee, das Wachs abzuschaben. Somit waren die Griechen vor dem Angriff der Perser informiert und konnten sich auf den Krieg, welchen sie übrigens gewonnen haben, vorbereiten. Die Übermittlung geheimer Nachrichten, bei der verborgen wird, dass überhaupt eine Botschaft existiert, heißt *Steganographie*.
Mit der Zeit wurden nun immer neuere und ausgefeiltere Methoden entwickelt, aber einen wirklichen Schutz boten sie auch nicht, da diese Verfahren mit der Zeit immer bekannter wurden und ein dritter an einem abgefangen Objekt einfach alle Möglichkeiten zur Entschleierung ausprobieren konnte.
Daher entstand zugleich auf die *Kryptographie*. Nicht die Existenz einer Botschaft zu verschleiern ist Ziel der Kryptographie, sondern ihren Sinn zu verbergen, und dies mittels eines Verfahren der Verschlüsselung. Um nun eine Botschaft für Außenstehende

unverständlich zu machen, muss sie nach einem bestimmten Verfahren gemixt werden, auf welches sich Sender und Empfänger vereinbart haben. Wenn der Empfänger nun die verschlüsselte Nachricht erhält, kann er das Verschlüsselungsverfahren einfach wieder rückwärts anwenden und erhält somit den Klartext. Der Vorteil an diesem Verfahren ist, dass Personen, die den Inhalt der Nachricht nicht erfahren sollen, die Botschaft nicht entschlüsseln können, solange sie nicht in Besitz des passenden Schlüssels sind.
Beide nun erwähnten Verfahren, Steganographie und Kryptographie, sind zwar zwei verschiedene Disziplinen, doch kann man sie trotzdem beide zusammen anwenden, d.h., dass man eine Nachricht sowohl verschlüsseln, als auch verbergen kann, was die Sicherheit der Botschaft wiederum erhöht.

3. Transposition und Substitution

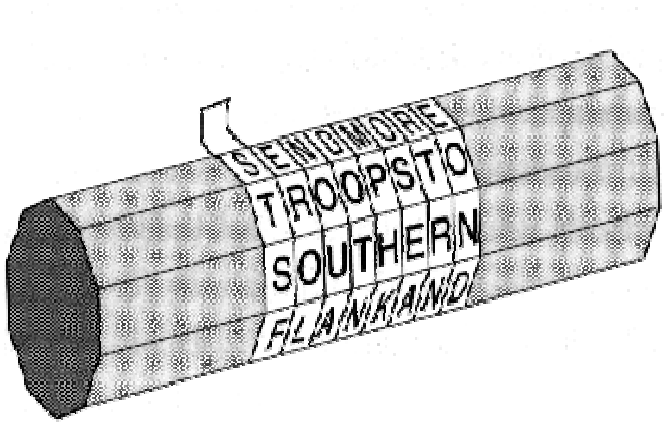
In der Kryptographie benutzt man hauptsächlich zwei Verfahren: die *Transposition* und die *Substitution*. Bei der Transposition werden die Buchstaben einer Botschaft einfach anders angeordnet, was nichts anderes als ein Anagramm ergibt. Doch bei sehr kurzen Mitteilungen ist dieses Verfahren äußerst unsicher, da die Möglichkeiten, die Buchstaben anders anzuordnen, relativ gering sind: n! (n-Fakuktät). Somit besitzt ein Wort mit 3 Buchstaben nur 3! = 3*2*1 = 6 Möglichkeiten. Wenn man aber nun Worte oder gar Sätze mit mehreren Buchstaben benutzt, dann steigt auch die Zahl der Anordnungsmöglichkeiten sehr schnell an. Bei 34 Buchstaben gibt es 34! = 34*33*32*...*3*2*1= 14 830 000 000 000 000 000 000 000 000 000 verschiedene Möglichkeiten. Allein an dieser astronomisch großen Zahl erkennt man, dass dieses Verfahren den Anschein erweckt, sehr sicher zu sein, da ein Mensch kaum in der Lage wäre, alle verschiedenen Möglichkeiten jemals auszuprobieren. Bei der Verschlüsselung durch die Transposition ist jedoch zu beachten, dass sich Sender und Empänger auf ein bestimmtes Verschlüsselungssystem einigen müssen, damit der Empfänger die Nachricht auch wieder entschlüsseln kann. Ein Beispiel zur Transposition wäre:

ICH HOFFE EUCH GEFAELLT DER TEXT

I H O F E C G F E L D R E T
C H F E U H E A L T E T X

IHOFECGFELDRET CHFEUHEALTETX

Wenn der Empfänger nun das vom Sender angewandte Verfahren rückwärts anwendet, dann kann er die Botschaft ohne Probleme entschlüsseln und lesen. Das erste militärische Kryptographie-Verfahren ist übrigens die Skytale, die schon im 5 Jhr. von den Spartanern entwickelt wurde.



Wie man auf dem Bild erkennen kann, wird ein Blatt um die Skytale gewickelt und die Nachricht dann auf dieses geschrieben. Wenn man nun das Blatt wieder abnimmt, dann stehen die Buchstaben untereinander und ergeben keinen Sinn. Sender und Empfänger müssen aber eine Skytale von gleicher Beschaffenheit besitzen, um die Nachrichten richtig ver- bzw. entschlüsseln zu können.

Eine Alternative zur Transposition ist die Substitution, bei der jeder Buchstabe des Klartextes durch einen anderen Buchstaben ersetzt wird. Dieses Verfahren ist gleichsam spiegelverkehrt zur Transposition. Bei der Transposition bleibt sich jeder Buchstabe gleich, doch er wechselt seinen Platz, während bei der Substitution jeder Buchstabe seine Gestalt wechselt, doch seinen Platz behält.

Einer der größten Feldherren, der diese Form der Verschlüsselung zu seinem Vorteil nutzte, war Julius Caesar:
Im Gallischen Krieg verwendete er die Substitution, um geheime Botschaften an seine Gefolgsleute zu versenden. Bei seiner Verschlüsselung ersetzte Caesar einfach jeden Buchstaben der Nachricht durch den Buchstaben, der drei Stellen weiter im Alphabet folgte. Darum wird dieses Verschlüsselungsverfahren auch Caesar-Verschiebung oder einfach nur Caesar genannt.
Wenn man nun die Buchstaben des Klartextalphabets und die des Geheimtextalphabets übereinander schreibt, dann lässt sich die Botschaft ganz einfach ver- bzw. entschlüsseln.
Ein Beispiel:

Klartextalphabet: a b c d e f g h i j k l m n o p q r s t u v w x y z
Geheimtextalphabet: d e f g h i j k l m n o p q r s t u v w x y z a b c

Klartext: eine verschluesselte botschaft
Geheimtext: hlqh yhuvfkohvvhowh erwvfhdiw

Caesar verwendete zwar nur eine Verschiebung, doch es liegt auf der Hand, dass es zwischen einer und 25 Stellen möglich ist, nur 25 verschiedene Geheimschriften zu erstellen. Aber wenn wir uns nicht darauf beschränken würden, das Alphabet zu verschieben und als Geheimtextalphabet beliebige Umstellungen des Klartextalphabets zulassen würden, dann könnten wir eine sehr viel größere Zahl unterschiedlicher Geheimschriften erzeugen und somit gibt es insgesamt 400 000 000 000 000 000 000 000 000 solcher Neuanordnungen.

4. Algorithmus und Schlüssel

Es lässt sich also leicht erkennen, dass jede einzelne Geheimschrift aus der Verknüpfung einer allgemeinen Verschlüsselungsmethode, dem *Algorithmus*, mit einem *Schlüssel*, der die Einzelheiten jeder bestimmten Verschlüsselung festlegt, entsteht. In dem vorigen Beispiel besteht der Algorithmus aus der Ersetzung jedes Buchstabens des Klartextalphabets durch einen Buchstaben eines Geheimtextalphabets, das für eine bestimmte Verschlüsselung verwendet wird. Somit kann der Gegner die Nachricht ruhig abfangen und den Algorithmus wissen, kann aber vermutlich nicht die Nachricht entschlüsseln, da er den passenden Schlüssel nicht kennt.

Sender		***Empfänger***
Schlüssel		Schlüssel
+	----> Algorithmus ----> Geheimtext ---->Algorithmus <----	+
Klartext		Klartext

Die Bedeutung des Schlüssels im Gegensatz zum Algortihmus ist ein bis heute unumstrittener Grundsatz der Kryptographie, dem der holländische Linguist Auguste Kerkhoffs von Nieuwenhof in seinem Buch "La Cryptographie Militaire" die endgültige Gestalt gab: „Die Sicherheit eines Kryptosystems darf nicht von der Geheimhaltung des Algorithmus abhängen. Die Sicherheit gründet sich nur auf die Geheimhaltung des Schlüssels. Doch so wichtig es ist, den Schlüssel sicher zu verwahren, so wird die Geheimhaltung nicht nur durch die sichere Aufbewahrung erhalten, sondern es ist auch eine sehr große Vielzahl an möglichen Schlüsseln nötig.“
So ist die Caesar-Verschlüsselung recht schwach, da es nur 25 mögliche Schlüssel gibt. So müsste der Gegner also nur die 25 Möglichkeiten ausprobieren und hätte dann den Klartext.

Wenn der Sender aber nun den allgemeineren Substitutions-Algorithmus verwendet, bei dem das Geheimtextalphabet eine beliebge Neuanordnung des Klartextalphabets sein kann, dann gibt es 400 000 000 00 000 000 000 000 000 mögliche Schlüssel.
Ein Beispiel:

Klartextalphabet: a b c d e f g h i j k l m n o p q r s t u v w x y z
Geheimtextalphabet: q a y w s x e d c r f v t g b z h n u j m i k o l p ö

Klartext ich bin nur ein text
Geheimtext cyd acg gim scgls

Ein großer Vorteil dieser Verschlüsselung ist es, dass sie relativ einfach anzuwenden ist und zudem ein so hohes Maß an "Sicherheit" bietet. So gibt es neben diesem Verfahren auch ein noch ein einfacheres, bei dem man sich nur das Schlüsselwort merken muss:

Klartextalphabet: a b c d e f g h i j k l m n o p q r s t u v w x y z
Geheimtextalphabet: c o m p u t e r s v w x y z a b d f g h i j k l n q

Alle Verfahren der Substitution gehören zu der *monoalphabetischen Verschlüsselung*. Somit müssen sich Sender bzw. Empfänger nur das Schlüsselwort "computer" merken und schon können sie das Geheimtextalphabet erstellen. Doch diese Vereinfachung reduziert die mögliche Anzahl von Schlüsseln und daraus resultiert wiederum eine Minderung der Sicherheit der Botschaft.

Dieses Verschlüsselungsverfahren stieg schnell zu hohem Bekannheitsgrad auf und wurde im ersten Jahrtausend zur Königin der Verschlüsselung, da es noch keine geeigneten Mittel gab, die Nachrichten schnell und einfach zu knacken. Doch auch hier tut sich schon ein Problem auf, welches im weiteren Verlauf der Geschichte der Kryptographie zu einem der schwierigsten und bedeutsamsten werden sollte, nämlich der sicheren Verteilung der Schlüssel.

5. Kryptoanalyse

Mit der Zeit entwickelte sich im Orient langsam das Gegenstück zur Kryptographie, nämlich die *Kryptoanalyse*. Die Kryptoanalyse ist die Wissenschaft von der Entschlüsselung ohne Kenntnis des Schlüssels. Während der Kryptograph neue Methoden der Verschlüsselung entwickelt, sucht der Kryptoanalytiker nach Schwächen in ebendiesen Verfahren, um in die geheimen Botschaften einzubrechen.

Und so gelang es auch den arabischen Kryptoanalytikern, ein Verfahren zu entwickeln, welches die monoalphabetische Verschlüsselung knacken konnte und sie somit wertlos machen sollte.
Die islamische Kultur war zu jener Zeit den Europäern in vielen Belängen weit überlegen und so konnten sich die Wissenschaften weiter entwickeln, welche für die Kryptoanalyse bedeutend sind. Das wären vor allem die Mathematik, die Statistik und die Sprachwissenschaften.
Die Kryptoanalytiker, die Ersten unter ihnen waren Religionsgelehrte, untersuchten viele Koranverse und bemerkten, dass einzelne Buchstaben häufiger vorkommen, als andere. Im Arabischen sind dies die Buchstaben "a" und "l". Diese zuerst unscheinbar wirkende Entdeckung war jedoch der Durchbruch in der Kryptoanalyse.

Eine Möglichkeit, eine verschlüsselte Botschaft zu entziffern, vorausgesetzt, wir kennen ihre Sprache, besteht darin, einen anderen Klartext in derselben Sprache zu finden, der lang genug ist, um ein oder zwei Blätter zu füllen, und dann zu zählen, wie oft jeder Buchstabe vorkommt. Wir nennen den häufigsten Buchstaben den "ersten", den zweithäuifigsten den "zweiten", den folgenden den "dritten" und so weiter, bis wir alle Buchstaben in der Klartextprobe durchgezählt haben.

Dann betrachten wir den Geheimtext, den wir entschlüsseln wollen, und ordnen auch seine Symbole. Wir finden das häufigste Symbol und geben ihm die Gestalt des "ersten" Buchstaben der Klartextprobe, das zweithäufigste Symbol wird zum "zweiten" Buchstaben, das dritthäufigste zum "dritten" Buchstaben und so weiter, bis wir alle Symbole des

Kryptogramms, das wir entschlüsseln wollen, auf diese Weise zugeordnet haben.
Dieses Verfahren wird als *Häufigkeitsanalyse* bezeichnet und mit der Zeit wurden ganze Tabellen erstellt, mit denen man genau bestimmen konnte, wie oft ein Buchstabe durchschnittlich vorkommt und somit war es ein leichtes, diese bis dahin so sichere monoalphabetiche Verschlüsselung zu knacken. Doch dadurch, dass eben nur die Durchschnittswerte in den Tabellen vorkommen, kann man diese Tabellen nicht als Schablonen zur Entzifferung verwenden. Es gibt im Deutschen und in anderen Sprachen bestimmte Redewendungen oder Zungenbrecher in denen bestimmte, ansonsten häufig vorkommene Wörter nicht auftauchen. Somit muss man dort aufpassen und zudem auch seinen Verstand einsetzen und fehlende Bausteine aus dem Sinn her einfügen.

Nachdem nun einige Grundlagen der Kryptographie und der Kryptoanalsyse erwähnt worden sind, möchte ich hier einen Schnitt vornehmen, da der Text ansonsten zu sehr ausarten würde. Die nachfolgenden Texte werden sinngemäß aufeinander aufbauen, sodass die Themen nicht einfach durch einander gewürfelt werden, sodass es noch ein wenig dauern könnte, bis wir bei den modernen Verschlüsselungsmethoden angekommen sind. Ich hoffe, dass es euch etwas Spaß gemacht hat, diesen Text zu lesen.

6. Vorschau

Der nächste Bericht wird weiter die Geschichte der Kryptologie erzählen, vorraussichtlich bis zur Entwicktlung der Enigma. Zudem wird auch das Problem der Schlüsselverteilung angedeutet. Sie können sich also auf einen etwas längeren Beitrag freuen.

7. Quellenangaben

Geheime Botschaften
Verschlüsselte Botschaften
Sicherheit und Kryptographie im Internet

-- Christoph 'debu' Wille

ARP-Spoofing

Erläuterung und C-Sourcecode

Einleitung

In diesem Dokument wird auf die Möglichkeit eingegangen durch eine Technik namens ARP-Spoofing in einem mit Switches ausgestattetem Netzwerk Daten anderer Netzwerkteilnehmer auszuspähen.

Hierbei wird betont, dass dies sowohl mit IPv4 als auch mit IPv6 und ähnlichem funktioniert, allerdings wird der Schwerpunkt auf IPv4 gesetzt, da dies das einzige Protokoll ist, welches heutzutage in Ethernet-Netzwerken standardmäßig genutzt wird.

Theorie

MAC Adressen

In einem Ethernet-Netzwerk besitzt jeder Netzwerkknoten eine Adresse, an dem man den Knoten identifiziert. Diese Adresse ist die Media Access Control Adresse – kurz MAC – und besteht aus sechs Bytes, die in Hex mit einem Doppelpunkt zwischen jedem Byte ausgeschrieben wird.

Beispiel einer MAC Adresse: 00:05:02:6E:2E:47

Über diese Adresse weiß ein Rechner, welche Pakete für ihn bestimmt sind, und wissen Switches, welche Pakete wohin weitergeschickt werden müssen. Es gibt eine spezielle Adresse, die Broadcast-Adresse. Wenn ein Switch ein Paket mit der Broadcast-Adresse als Empfängeradresse empfängt, schickt es das Paket an alle Ports weiter. Wenn ein Rechner ein Paket mit der Broadcast-Adresse als Empfängeradresse empfängt, nimmt er an, dass es für ihn bestimmt ist.

Broadcast MAC Adresse: FF:FF:FF:FF:FF:FF

Um die MAC Adresse einer Netzwerkkarte eines Rechners herauszufinden, ruft man unter Linux und Konsorten den Befehl *ifconfig* in der Konsole auf. Unter Windows ruft man den Befehl *ipconfig /all* auf.

IPv4 Adressen

In einem Internet Protocol Version 4 – kurz IPv4 – Netzwerk, was fast alle heutzutagigen Netzwerke sind, werden allerdings keine MAC Adressen verwendet, um einen Rechner zu identifizieren, sondern – wie der Name schon sagt – vier Byte lange IPv4 Adressen. Diese Adressen werden in Dezimalform mit einem Punkt zwischen jedem Byte ausgeschrieben.

Beispiel einer IPv4 Adresse: 192.168.0.1

Auch bei IPv4 Netzwerken gibt es eine Broadcast Adresse, wo auch die Pakete bei allen Rechnern ankommen.

Broadcast IPv4 Adresse: 255.255.255.255

Der Kleber: ARP

Das Address Resolution Protocol – kurz ARP – ist verantwortlich für die Umrechnung zwischen den MAC und IPv4 Adressen.

Wenn ein Rechner in einem Ethernet-Netzwerk auf einen anderen Rechner zugreifen will, sagen wir mal 192.168.0.4 will auf 192.168.0.1 zugreifen, dann schickt er, falls er die MAC Adresse

des Rechners 192.168.0.1 noch nicht im Speicher hat, ein ARP Request an die Broadcast MAC Adresse. Falls 192.168.0.4 im Netzwerk ist, wird er nun in einem ARP Reply seine MAC Adresse verraten, die 192.168.0.1 benötigt, um die Daten direkt an 192.168.0.4 zu schicken.

ARP Request
00:05:02:6E:2E:47 an FF:FF:FF:FF:FF:FF
Welche MAC Adresse hat 192.168.0.4?

Ein ARP Request wird an das gesamte Netzwerk gesendet.

192.168.0.4 schickt ein ARP Reply an 192.168.0.1.

ARP Reply
00:0C:6E:2D:DE:18 an 00:05:02:6E:2E:47
192.168.0.4 hat die Adresse 00:0C:6E:2D:DE:18.

Falls 192.168.0.1 den ARP Reply innerhalb einer bestimmten Zeit empfängt, kann er anfangen die IPv4 Pakete zu schicken, da er 192.168.0.4s MAC Adresse kennt und somit andere Rechner auf dem Netzwerk die Daten nicht empfangen können (vorausgesetzt Switches werden genutzt). Falls der Reply innerhalb der Zeit nicht empfangen wurde, glaubt 192.168.0.1 zu wissen, dass der Recher 192.168.0.4 nicht in dem Netzwerk verfügbar ist und zeigt dem Nutzer eine Fehlermeldung oder ähnliches.

ARP-Spoofing: Definition

ARP-Spoofing bezeichnet das Versenden von gefälschten ARP Reply Paketen um vorzugauklen, dass ein IPv4 Knoten sich bei einem anderen Rechner befindet als in Wirklichkeit. Dies funktioniert deshalb, weil ARP ein zustandloses Protokoll ist, das heißt, dass ARP Reply Pakete nicht an ARP Request Pakete gebunden sind. Dies bedeutet aber auch, dass Netzwerkknoten jedes ARP Response Paket akzeptiert, welches es empfängt, und die neue MAC Adresse eines IP-Knotens speichert. Somit schickt er alle zukünftigen IP-Pakete, die für den IP-Knoten bestimmt sind, an die neue MAC Adresse.

Dies lässt sich in folgenden Weisen ausnutzen:

- Denial of Service

  Durch Schicken eines ARP Reply Paketes, welches eine IP-Adresse mit einer nicht-existierenden MAC Adresse verknüpft, an die Broadcast MAC Adresse lässt sich auf einfachster Weise sicherstellen, dass ein Rechner keine IPv4 Pakete mehr empfängt. Somit ist der Rechner von der Außenwelt abgeschottet. Wenn man dadurch den Router/Gateway des Netzwerks abschottet, kann kein Rechner mehr an die Aussenwelt zugreifen.

- Man-in-the-Middle Attack

  Hier verknüpft man die IP-Adresse des Rechner Nr. 1 auf Rechner Nr. 2 mit seiner eigenen MAC Adresse und umgekehrt. Dadurch bekommt man selbst die Pakete, die eigentlich von Rechner 1 an Rechner 2 gedacht waren und umgekehrt. Nun kann man die Pakete auswerten und danach an die richtigen Empfänger weiterschicken.

- Broadcast Attack

  Hier verknüpft man sowohl die IP-Adresse des Rechner Nr. 1 auf Rechner Nr. 2 als auch die IP-Adresse des Rechner Nr. 2 auf Rechner Nr. 1 mit der Broadcast Adresse. Hierdurch empfängt jeder Rechner alle Daten der beiden Rechner, unter anderem auch der Angreifer-Knoten. Dieser kann die Daten nun auswerten und muss die Pakete auch nicht weiterverschicken, da die eh auch an den richtigen Empfängern ankommen.

- Connection Hijacking

  Diese Technik passt nicht ganz unter dem Stichwort des ARP-Spoofing. Hier wird einfach die

gesamte Identität eines Rechners im Netz übernommen, das heißt man übernimmt sowohl die MAC als auch die IP Adresse. Dies ist unter Linux mit dem eingebauten Programm ipconfig ohne Aufwand möglich. Sehr interessant ist dies in öffentlichen WLAN Hotspots.

Bei der Denial of Service und der Broadcast Attacke lässt sich die Attacke durch geeignete Programme auf allen Rechnern jederzeit identifizieren, bei der Man-in-the-Middle Attack aber nur bedingt. Es gibt heutzutage allerdings dank des Designs des Ethernet-Netzwerks keine Vorkehrungen, die getroffen werden können, um sicherzugehen, dass ARP-Spoofing nicht möglich ist.

ARP-Spoofing: Praxis

Using Programs

Der König aller ARP-Spoofing Programme ist Ettercap, welches das Abhören in geswitchten Netzwerken durch Automation zu einer Kinderaufgabe vereinfacht. Nach Eingabe eines Source- und eines Destination-Knotens schickt Ettercap die gespooften ARP-Replys an die Rechner und protokolliert danach alle TCP Streams und sonstige Pakete.

Mit dem Programm arpspoof lässt sich einzelne gespoofte Pakete verschicken. Die Pakete lassen sich nun mit einem Programm wie Ethereal protokollieren. Im Falle eines Man-in-the-Middle Attackes lässt sich zum Forwarding der Pakete an den richtigen Rechner das eingebaute Paketforwarding nutzen.

Paketforwarding unter Linux einschalten: echo 1 > /proc/sys/net/ipv4/ip_forward

For Real Programmers

Wer sich mit Programmiersprachen (C/C++) gut auskennt, kann sich auch leicht selbst Spoofing-/Sniffingprogramme schreiben.

Um auf Ethernet Protokollebene Pakete schicken und empfangen zu können, benötigt es eine Third Party Bibliothek. Bewährt hat sich pcap, welches für Windows Platformen (*WinPcap*) und UNIX-basierte Platformen insklusive Linux (*libpcap*) existiert. Bei beiden ist das Interface dasselbe, was Cross-Platform Programmierung vereinfacht.

Ein gespooftes ARP-Paket zu schicken ist nicht schwer. Ein ARP-Paket beginnt erst mal mit einem Ethernet Frame II Header, auf welches das ARP-Paket aufbaut:

6 Bytes – Destination MAC Address
6 Bytes – Source MAC Address
2 Bytes – Protokoll-ID (ARP: 0x0806)

Nach diesen 14 Bytes kommt das ARP-Paket an die Reihe:

2 Bytes – Typ der Hardware (Ethernet: 1)
2 Bytes – Protokoll (IPv4: 0x0800)
1 Byte – Größe der MAC Adresse (6)
1 Byte – Größe der IPv4 Adresse (4)
6 Bytes – Source MAC Address
4 Bytes – Source IPv4 Address
6 Bytes – Destination MAC Address
4 Bytes – Destination IPv4 Address

Die Source und Destination MAC Adressen des Ethernet Frame II Headers und des ARP Paketes sollten identisch sein, da einige Netzwerkknoten sonst die ARP Pakete verwerfen.

Nun müssen diese 40 Bytes nur noch an pcap_sendpacket geschickt werden.

Ich habe dazu selbst ein Programm erstellt, welches ARP-Spoofing ausübt. Zu finden ist der Quelltext unter http://spaxid.it-helpnet.de/files/1/arp-spoofing/

ARP-Spoofing: Abwehr

Wie schon vorher erwähnt lässt sich ARP-Spoofing leider nicht abwehren, sondern nur feststellen. Dies ist auch nur möglich, wenn der Rechner, auf dem das Programm zur Feststellung läuft, die gefälschten ARP Pakete auch empfängt (dies ist zum Beispiel bei der Man-in-the-Middle Attacke nicht immer der Fall).

Hier empfielt sich der Einsatz des Programmes arpwatch, welches alle vorkommenden ARP Pakete protokolliert und bei merkwürdigen Veränderungen den Administrator des Netzwerkes per E-Mail alarmiert.

ARP-Spoofing lässt sich nur vermeiden, indem man die Benutzung von ARP ganz umgeht. Hierzu gibt man in jedem Rechner eine statische MAC<->IP Zuordnungstabelle ein, sodass ARP gar nicht benötigt wird.

Schlusswort

ARP-Spoofing ist eine Technik, die nicht jeder kennt. Der, der sie kennt, kann alles innerhalb eines Netzwerkes abhören, auch was nicht für ihn bestimmt ist. Dies ist leider per Design so, ohne eine Umstellung der Ethernet-Technologie wird sich das nicht ändern lassen.

Ich hoffe, dass ich Ihnen ein paar Basisinformationen liefern konnte. Für weitergehende Informationen bietet sich Google, oder Altavista für diejenigen, die Google heutzutage eher meiden, an. Auch bin ich bereit bei Anfragen auszuhelfen.

-- Thomas 'RoCKdaFrogg' Rogg

Dies ist das Ende der ersten Ausgabe des *Spaxid* eZines. Wir hoffen, es hat Ihnen gefallen und Sie nehmen etwas von den Informationen mit, die wir versucht haben, Ihnen zu präsentieren.

Bei Kritik, Fragen oder Verbesserungsvorschlägen kontaktieren Sie bitte die jeweiligen Autoren per E-Mail. Für Fragen können Sie außerdem das Forum auf www.computer-support.org nutzen. E-Mail Adressen, Bannergrafiken oder die folgenden Ausgaben von *Spaxid* finden Sie auf http://spaxid.it-helpnet.de