Firewalls und Proxies von Anwendern ausgetrickst

Firewalls und Proxies von Anwendern ausgetrickst

[expand title="mehr lesen…"]

Firewalls und Proxies von Anwendern ausgetrickst

Durchgegraben

Thomas Ronzon

Nicht nur Netzwerkprofis kennen Firewalls. Inzwischen wissen auch Anwender, wie sie die vermeintlich sichere Firmenmauer untergraben können.

Zu den Aufgaben von Firewalls gehört es, schädliche Dateien aus dem Netz fernzuhalten. Gerade hier kollidieren die Interessen von Administratoren und Anwendern am heftigsten. Die einen suchen nach Wegen, unwillkommene Exemplare zuverlässig herauszufiltern, die anderen danach, die daraus resultierenden Einschränkungen zu umgehen.

Da Schädlinge ihre Reise ins Firmennetz gern per E-Mail antreten, gehört das Prüfen von Mail-Anhängen zum Tagesgeschäft. Prüft der Mailserver jedoch nur die Dateiendungen, muss der Absender etwa ein gepacktes Archiv *datei.zip* nur mit dem Namen *datei.txt* versehen. Der Empfänger

braucht die Datei lediglich wieder umzubenennen.

Gerade Administratoren mit Windows-Hintergrund scheinen für diesen Fehler anfällig zu sein. Deutlich zuverlässiger lässt sich erkennen, dass es sich etwa um ein mit dem ZIP-Algorithmus gepacktes Archiv handelt, wenn man den MIME-Type (Multipurpose Internet Mail Extensions) – quasi deren Fingerabdruck – bestimmt.

Ähnlich arbeiten Shell-Skripte auf unixoiden Betriebssystemen unter Zuhilfenahme des Befehls *file*, der den Dateityp anhand sogenannter Magic Patterns bestimmt. Auch für andere Sprachen stehen solche Werkzeuge bereit. Java-Entwickler etwa können auf das Java-Paket SimpleMagic zurückgreifen. In der Windows-Welt dagegen sind derartige Befehle unbekannt.

Doch auch die Prüfung des MIME-Typs lässt sich ganz einfach umgehen. Es genügt, die nicht druckbaren Zeichen einer binären Datei mit dem Befehl *uuencode datei1.zip datei1.zip > datei1.uue* oder einem grafischen Derivat für Windows in druckbare ASCII-Zeichen zu wandeln und mit *uudecode datei.uue* wiederherzustellen. Mit dieser Technik übertrug man in den Kindertagen des Internets binäre Dateien per E-Mail, da Mails nur eine 7-Bit-Kodierung besaßen. *uuencode* teilt drei Bytes der Binärdatei auf vier 6-Bit-Werte auf und ordnet ihnen druckbare ASCII-Zeichen zu.

Ähnlich wie Filter von E-Mail-Anhängen verhalten sich Proxies, wenn Anwender *.exe*-Dateien per Browser herunterladen wollen. Das Download-Ziel von *www.meinedomain.de/tool.exe* erkennen viele Proxies anhand der Endung als ausführbare Datei und sperren sie, nicht dagegen *www.meinedomain.de/tool.exe?*. Diese URL gaukelt Proxies, die nur die letzten drei Zeichen prüfen, die Endung *xe?* vor, während der Webserver das *?* als regulären Ausdruck erkennt, der für ein beliebiges, also auch kein Zeichen stehen kann. In der Regel verwirft er das Zeichen einfach und liefert die richtige Datei aus.

Umbenennen und teilen

Ein anderes Mittel vieler Administratoren besteht darin, dass sie nur Mails bis zu einer bestimmten Größe durchlassen. Damit wollen sie erreichen, dass große und damit in den Augen mancher Administratoren gefährliche Dateien nicht ins eigene Netz gelangen. In Zeiten teuren Speicherplatzes war dies sicherlich auch ein Mittel, um der Verschwendung von Ressourcen vorzubeugen.

Bei Anwendern kursieren deshalb althergebrachte Programme wie HJSplit, das bis in die 16-Bit-Zeit zurückreicht. Grafische Varianten sind für sämtliche Windows-Versionen, Linux, Mac OS und BSD verfügbar. Versierte Linux-Anwender zerlegen die Datei schlicht mit *split –b <size> daten.xls split-daten.xls*, wobei *<size>* die Größe der Teildateien in Byte angibt. Mit *cat split-daten.xls.* > daten.xls* setzen sie die Datei wieder zusammen.

So gut wie jeder Mitarbeiter hat neben seinem beruflichen Postfach einen privaten Account. Die Regeln dort sind meinst weniger restriktiv als bei Firmen-Accounts. Ein beliebtes Mittel, Dateien zu übertragen, ist das Senden an private Accounts. Der Kollege muss nun nur noch die E-Mail abholen und den Inhalt auf seinem Rechner speichern.

Theoretisch ließe sich der Zugang zum Webmailer sperren – aber: Webmailer gibt es wie Sand am Meer. Beliebt ist auch, die E-Mail mit dem Smartphone abzurufen. Dazu verbindet der Smartphone-Besitzer einen Rechner per Tethering, also über USB, WLAN oder Bluetooth, mit seinem Gerät und nutzt es als Hotspot.

Von Rechner zu Rechner

In vielen Firmen gehört das Sperren der USB-Ports zu den Standardmaßnahmen, vor allem, wenn sie nicht nur verhindern wollen, dass Schadsoftware auf den PC gelangt, sondern auch,

dass Daten das Firmennetz verlassen. Auch das lässt sich leicht aushebeln.

Ein Anwender muss dazu nur auf einem anderen Rechner einen VNC-Server starten. Ist dort der File-Transfer aktiviert, kann er von seinem PC per VNC-Client darauf zugreifen. Noch brisanter ist die Lage bei Werkzeugen wie TeamViewer, da sie die Dateien nicht direkt an den zweiten Rechner übertragen, sondern auf einem Server zwischenspeichern. Das geschieht zwar verschlüsselt, bietet jedoch Angriffsflächen.

Noch einfacher ist der Weg über Filehoster. Auch hier stellt das Ablegen der Daten auf Servern des jeweiligen Dienstleisters wie Dropbox, Hightail oder Skype ein nicht zu unterschätzendes Sicherheitsrisiko dar. Das Sperren der Domains hilft wenig, da es mittlerweile so viele Filehoster gibt, dass es einem ewigen Katz-und-Maus-Spiel gleichen würde.

Durch einen Tunnel zum SSH-Server und von dort weiter zum Webserver – auf diese Weise können versierte Anwender etwa die Sperren zu bestimmten Websites unterlaufen.

Weitere Optionen, firmenseitige Restriktionen zu umgehen, stehen fortgeschrittenen Anwendern offen. Eine davon ist das Tunneln, da für die Firewall nur das Tunnelprotokoll sichtbar ist. Ist etwa der Zugriff auf *sshserver* per SSH erlaubt, möchte der Anwender aber auf einen Webserver zugreifen, tunnelt er die HTTP-Anfrage per SSH. Windows-Benutzer greifen hier meist zu *putty*, Linux-Anwender brauchen auf ihrem Client nur die Datei *~/.ssh/config* anzulegen und mit dem folgenden Inhalt zu füllen:

```
Host <sshserver>
    HostName <sshserver.domain.de>
    User <user>
    LocalForward 20000
 <webserver>:80
```

Das verkürzt den Befehl *ssh –L 20000:<webserver>:80*

<sshserver.domai.de> zum Öffnen des Tunnels auf *ssh <sshserver>*. Eine Browseranfrage an *localhost* Port 20000 mit *http://localhost:20000* tunnelt die SSH einfach bis zum *sshserver* und leitet sie von dort an den Webserver Port 80 weiter (siehe Abbildung).

Alternativ kann man auf dem eigenen Arbeitsplatz einen Webserver starten. Windows-Benutzer haben etwa mit Abyss, HFS (HTTP File Server), jibble oder dem Mini-Webserver 2010 einige schlanke Apache-Alternativen zur Auswahl, Linux-Anwender können aus dem Vollen schöpfen. Eine Ad-hoc-Variante in Form eines Bash-Kommandos hat der Entwickler Răzvan Tudorică vor einigen Jahren ins Netz gestellt:

```
while true; do { echo -e
 'HTTP/1.1 200 OK\r\n';
 cat </dir/datei>; } |
 nc -l 8888; done
```

Führt ein Anwender das Kommando auf seinem Linux-System *<linuxhost>* aus – so *netcat* vorhanden ist –, kann jeder andere, für den *<linuxhost>* erreichbar ist, mit seinem Browser über die URL *http://linuxhost:8888* auf die Datei zugreifen. Da *netcat* oder *nc* die Informationen zum Zugriff auf *stdout* ausgibt, kann der Anwender dort sehen, wann er die *while*-Schleife beenden kann.

Fazit

Ein Administrator verfügt über die Mittel, den gesamten Datenverkehr zu reglementieren und das Firmennetz abzuschotten. Ist der Betrieb von der Kommunikation mit anderen abhängig, etwa von der Zusammenarbeit mit anderen Unternehmen, ist das jedoch praxisfremd.

Besser wäre es, den Mitarbeitern geeignete Austauschwege und Zugänge zur Verfügung zu stellen. Wenn zudem die Mitarbeiter für die Vertraulichkeit der Firmendaten und die Gefahren sensibilisiert sind, sollten gefährliche Tricks wie das

Ablegen von Firmendaten in privaten Dropbox-Accounts der Vergangenheit angehören. Gegen böswillige Spionage von innen ist allerdings kein Kraut gewachsen. (sun) Thomas Ronzon arbeitet als Projektleiter und Senior Softwareentwickler bei der w3logistics AG.

Literatur

- [1] Thomas Ronzon; Das Licht am Ende des Tunnels; JavaSpektrum 1/2013

Durchgegraben

- SimpleMagic – Simple Magic Content and Mime Type Detection Java Package
  1
- Mark’s Lab; UUDWin
  2
- HJ-Split Freeware multi-platform file splitters
  3
- Ultra VNC
  4
- putty
  5
- Blog von Razvan Tudorica
  6
- Heise Netze; Dusan Zivadinovic; Webserver als Shell-Einzeiler
  7
- Abyss Web Server
  8
- HFS – Http File Server
  9
- jibble
  10
- Mini-Webserver 2010
  11

[/expand]