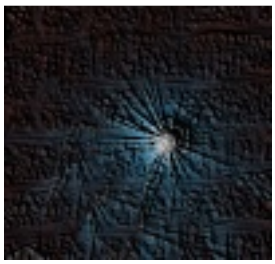
Firewalking - what / why / how
by Marco Gerber



- *Mit was beschäftigt sich firewalking*
- *Paketaufbau*
- *Tools*
- *Was man dagegen tun kann*

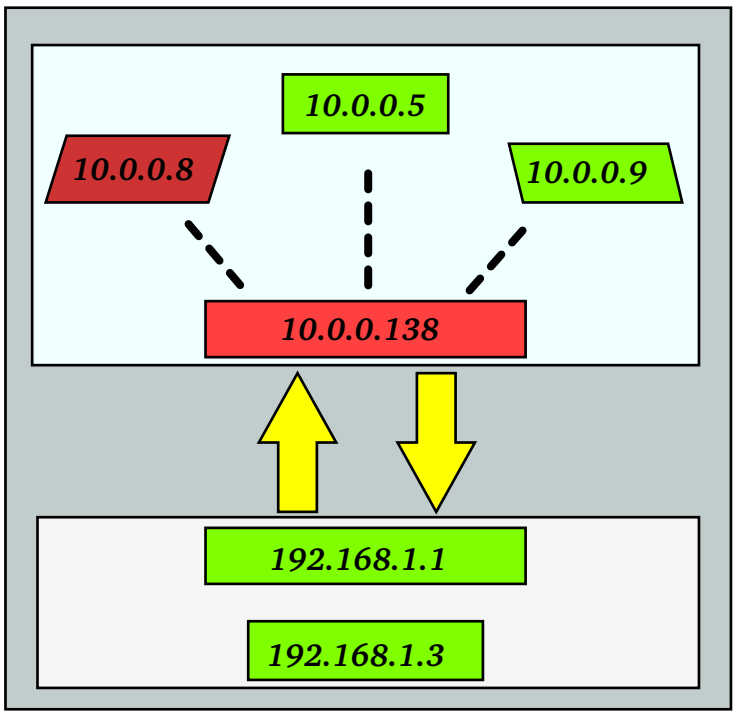
Mit was beschäftigt sich firewalking

Beim firewalking geht es darum, mittels gezielten Netzwerkpaketen einen Pfad durch eine Firewall zu finden. Dabei bedient man sich bekannten und unbekannten Fehlkonfigurationen auf Rules / ACLs.

Zuerst einmal muss man sich bewusst sein, dass ein System immer nur so sicher ist, wie ein Angreifer dumm.
Und genau dem liegt das Problem der Sicherheit von Firewallsystemen zu Grunde.
Grundsätzlich ist ein grosses Missverständnis im Umlauf, welches darauf beruht, ein System dadurch abzusichern, indem ICMP ECHO REQUEST Pakete abgefangen werden, um so die Existenz eines Rechners zu verbergen.
Viele Programme, welche sich zum flächendeckenden Scannen von Netzwerken eignen, ermitteln das Vorhandensein eines Rechners durch das Senden eines solchen "ping" Paketes. Dabei entscheiden diese aufgrund der erhaltenen Antwort, ob ein Rechner existiert oder nicht. Abgesehen davon, dass es sowieso der Ethik widerspricht, benutzen solche Programme für diese Art des Netzwerkscannings sowieso meist nur sogenannte Skript Kiddies.
Sie stellen in der Regel keine grosse Gefahr dar.

Beim firewalking geht es aber um eine andere Art des hackens (zur Begriffserläuterung siehe http://www.catb.org/~esr/faqs/hacker-howto.html). Es geht darum, gezielt durch die Kenntnis der Existenz eines Rechners die Firewallkonfiguration zu diesem System zu testen.

Für unsere Zwecke hier bauen wir ein virtuelles Netzwerk wie in der nebenstehenden Abbildung auf.
Es dient den grundsätzlichen Überlegungen für die nachfolgenden Erläuterungen.



[Ziel-Netz 255.0.0.0]
10.0.0.5 : Rechner
10.0.0.8 : Ziel-Rechner
10.0.0.9 : Rechner
10.0.0.138 : Router + Firewall

[Quell-Netz 255.255.255.0]
192.168.1.1 : Router
192.168.1.3 : Rechner, von welchem aus der Test erfolgt.

Unsere Voraussetzungen zum firewalking sind nun die folgenden:
· Wir wissen über die Existenz des Zielrechner 10.0.0.8 bescheid.

Das Ziel ist folgendes:
· Gibt es einen Fehler in der Firewallkonfiguration zum Zugang zum Zielrechner, welche sich für Datentransfer eignet?
· Für Datentransfer eignen sich Protokolle, welche auf IP aufsetzen (TCP/UDP usw)

Eine traceroute Ausgabe von 192.168.1.3 zum Zielrechner wird dann in etwa wie folgt aussehen:

[theoretische Ausgabe]

1	*192.168.1.1*	*2.400ms*	*2.500ms*	*2.550ms*
2	*10.0.0.138*	*2.550ms*	*2.600ms*	*2.670ms*
3	*10.0.0.8*	*3.010ms*	*3.000ms*	*3.120ms*

[reale Ausgabe]

1	*192.168.1.1*	*2.400ms*	*2.500ms*	*2.550ms*
2		*	*	*
3		*	*	*

Reale Ausgabe bedeutet, dass die Firewall unsere Tracepakete abgefangen hat, und nach einem Timeout Traceroute entscheidet, dass das Ziel nicht ansprechbar ist.

Wir wissen, dass Traceroute standardmässig so arbeitet, dass nach jedem gesendeten Paket der Zielport (UDP) um 1 inkrementiert wird. Der Startport ist auf 33435 festgelegt. Er lässt sich anpassen.

Die Route zum Ziel ist uns noch nicht bekannt.

Paketaufbau:

Es geht nun darum, einen Port zu finden, welcher für eine Datenübertragung von der Firewall zugelassen ist. Hier bieten sich nun die Möglichkeiten, die Startports selbst zu wählen. Der TTL Wert im IP Header kann genau an die abzufragende Nummer gesetzt werden.
Jedoch wird zunächst mit ICMP versucht, die Route komplett aufzuzeichnen. Dies liefert uns wertvolle Informationen darüber, wie weit es bis zum Ziel noch ist.
Wir weisen also Traceroute an, ICMP ECHO REQUEST Pakete zu senden.
Genau gleich wie zuvor, werden die Router zum Ziel fortlaufend den TTL 0 Wert mit einem TIME EXCEEDED IN-TRANSIT anzeigen.
Der Vorteil bei dieser Methode liegt darin, dass wir einen ECHO REQUEST senden, welcher von den meisten Routern im Gegensatz zu einem UDP Paket passierbar ist.

traceroute -I 10.0.0.8 liefert dann:

1	*192.168.1.1*	*2.400ms*	*2.500ms*	*2.550ms*
2	*10.0.0.138*	*2.550ms*	*2.600ms*	*2.670ms*
3	*10.0.0.8*	*3.010ms*	*3.000ms*	*3.120ms*

Nun kennen wir den Router, welcher die UDP Pakete geblockt hat. Auf diesem wird eine Regel sein, welche UDP Traffic auf dem gewählten Port unterbindet.

Ab hier wissen wir folgendes:
· Der Router besitzt eine Firewallregel (wobei Firewall hier breit gefasst ist)
· Hinter dem Router befindet sich im gleichen Subnet unser Zielrechner.

Was wir aber noch nicht wissen ist:
· Ist der antwortende Rechner auch der, für welchen wir ihn halten
· Um was für eine Art Netzwerk handelt es sich, in welchem sich unser Zielrechner befindet.

Kann man nun in Erfahrung brigen, um was für eine Art Netzwerk es sich handelt, kann man die mit grosser Sicherheit zugelassenen Dienste stark eingrenzen.
Auch hier hilft uns ICMP weiter. Nämlich mit dem Code 17 (ADDRESS_MASK_REQUEST). Dieser Code weist einen Router an, die Subnetmask seines Netzwerkes zurück zu geben. Hierzu kann man sich dem Tool hping bedienen, welches die Möglichkeit zum gezielten Erstellen eines Paketes liefert:

hping2 10.0.0.138 --icmptype 17 -c 1 2>&1 | grep "mask" (und oder 10.0.0.8)
-> ICMP address mask: icmpam=255.0.0.0

Es muss sich also um ein sehr grosses Netzwerk handeln, welches im Privatbereich benutzt wird (wissen wir eigentlich schon lange, aber ich verwende hierfür bewusst private Netzwerke - Anm. es handelt sich um das Netzwerk auf Seiten des Aufrufers). So ist die Wahrscheinlichkeit also sehr gross, dass sicher DNS zugelassen ist. Ebenfalls vermutlich noch ein syslogd, welcher von aussen her angesprochen werden kann. Server werden bewusst keine betrachtet, da diese für unsere Belangen nicht von Interesse sind. Ebenfalls ist die Umgebung von Servern meist gründlicher abgesichert als der Rest.

Mit den gewonnenen Daten lässt sich nun ein zweiter Anlauf mit traceroute starten, welcher diese Erkenntnisse nutzt.

traceroute 10.0.0.8 -p 53 -f 2 -n

*2 10.0.0.138 2.550ms * **

Damit haben wir einen Weg gefunden, welcher für einen Datenaustausch brauchbar ist.

Eingesetzte Tools:

Im Grunde reichen ein paar wenige Tools, welche uns Netzwerkpakete generieren.
Ein dafür geeignetes findet man unter dem Namen "hping".

- *hping*
- *traceroute*
- *normale UNIX Tools*

Was man dagegen tun kann:

Grundsätzlich eine Einschränkung des Zugriffes durch die oben beschriebenen Möglichkeiten, in dem man hauptsächlich den UDP und ICMP Verkehr gezielter regelt. Dies wird oft vernachlässigt, weil UDP nicht so sehr verbreitet ist in den bekannteren Angriffsverfahren auf den Netzwerkstack und Netzwerke im allgemeinen.

Ein ICMP_NETWORK_UNREACHABLE (Messagetype 2 Code 11) kann das Scannen mit den gezeigten ICMP Paketen einschränken.

Auf Linux Systemen eignet sich iptables als Frontend für netfilter zur Konfiguration geeigneter Rules:

```
iptables -A INPUT -i <DEVICE> -d <NETWORK> -p icmp --icmp-type \
address-mask-request -j REJECT --reject-with icmp-net-unreachable

iptables -A INPUT -i <DEVICE> -d <NETWORK> -p icmp --icmp-type \
echo-request -j REJECT --reject-with icmp-net-unreachable
```

Autor:

Marco Gerber ist Student der Informatik und beschäftigt sich seit nun mehr als 5 Jahren mit Linux/UNIX und Computersicherheit.

http://squeez.gurit.net