Don't Get Owned at DEF CON

Single Packet Authorization and SSH Tunneling

DEF CON 22

Jay Beale
InGuardians

Single Packet Authorization

“Single packet authorization” is an advancement on the “port knocking” concept.

I can set my firewall to allow no incoming connection attempt packets, while allowing myself to SSH in from any IP.

Portscans don’t show an open port.

Dynamically Open a Port

We can send a packet to the firewall to ask it to open up the SSH port for 10 seconds to only our IP.

The recipient port doesn't have to be open - the firewall can get the packet from either pcap (sniffing) or Netfilter's ulogd.

The recipient machine doesn't even need to exist!

fwknop

This concept has been implemented a number of times over the years. This talk looks at one of the longstanding implementations that is most actively maintained: **fwknop**, by Michael Rash.

Rash wrote the No Starch Press Linux firewalls book.

Fwknop, the Firewall Knock Operator, grew out of a port knocking solution.

http://www.cipherdyne.com/fwknop/

fwknop Basics

In its default configuration, the client, fwknop, sends a packet to UDP port 62201 that contains authentication information.

The server, fwknopd, is configured to open specific (or all) ports for each user. I configure it to open my SSH port, tcp/19851.

Why might I change my SSH port?

Platforms

Server runs on:

Linux	iptables
Mac OS X	ipfw
FreeBSD	ipfw
OpenBSD	PF

Client runs on the above operating systems, and also on Windows, Android and jailbroken iOS.

fwknop Packet Contains

Random Data
Username
Timestamp
Action
Access details (ip, port)
Hash of data (integrity checking)

Fighting Replay Attacks

Replay has always been a weakness in port-knocking.

Fwknop uses random data to keep packets changing and uses timestamps.

Fwknopd stores hashes of previous packets, not allowing a repeat packet.

fwknop Authentication

The default configuration uses a shared secret to encrypt the payload.

You can also use GPG (free PGP) instead:

Client signs to authenticate and encrypts with the server's public key.

Setup

Let's set this up for a Linux system that has sshd listening, but has a default-deny firewall rule for TCP port 19851.

fwknop Installation

First, compile and install.

```
# tar -xzvf fwknop-*.tar.bz2
# cd fwknop-*
# ./configure && make && make install
```

(You made need to install gcc or the libpcap-devel RPMs via yum)

Configure Network Interface

In /etc/fwknop/fwknopd.conf, add a line with the network interface:

```
PCAP_INTF                       eth0;
```

(If you built from source, this file is likely in: /usr/local/etc instead of /etc.)

Configure Server Sniffing

If packets are addressed to firewall itself, edit the ENABLE_PCAP_PROMISC line in /etc/fwknop/fwknopd.conf:

```
# Set this to no if the knocks will always be destined for
   this system.
ENABLE_PCAP_PROMISC          N;
```

Create Keys on the Client

```
$ fwknop -A tcp/19851 -a 192.168.1.1 -D
  spaserver.domain.com --key-gen --use-hmac --
  save-rc-stanza
[+] Wrote Rijndael and HMAC keys to rc file: /
  home/jay/.fwknoprc

$ grep KEY /home/jay/.fwknoprc
KEY_BASE64          Sz16sS8BTUsg=
HMAC_KEY_BASE64     1211H3NM5Q==
```

Place Keys on the Server

Create a shared secret entry in /etc/fwknop/access.conf:

```
SOURCE                          ANY
REQUIRE_SOURCE_ADDRESS          Y
KEY_BASE64                      Sz16sS8BTUsg=
HMAC_KEY_BASE64                 1211H3NM5Q==
```

Start Server and Test

server # fwknopd

client # ssh spaserver.domain.com

<hangs>

client # fwknop -n spaserver.domain.com

client # ssh spaserver.domain.com

client # iptables-save | grep FWKNOP

-A FWKNOP_INPUT -s 1.1.1.1/32 -p tcp -m tcp --dport 22 -m comment --comment "_exp_1406882245" -j ACCEPT

fwknop Effect

This opens port 19851 to syn packets from our IP for 10 seconds.

Rules are added to and removed from the Netfilter FWKNOP_INPUT chain.

Watch /var/log/messages.

fwknop and GPG

You can use the power of asymmetric encryption for environments with more than one user.

This requires GPG.

Exchange GPG Keys

http://www.cipherdyne.com/fwknop/docs/gpghowto.html)

```
Client $ gpg --gen-key
Client $ gpg -a --export <keyID> > client.asc
Client $ scp client.asc root@server:.
Server # gpg --import client.asc
Server # gpg --edit-key <ClientkeyID>
>sign
>save
Server # gpg --gen-key
Server # gpg -a --export <keyID> > server.asc
Server # scp server.asc user@client:.
Client $ gpg --import server.asc
Client $ gpg --edit-key <ServerkeyID>
>sign
>save
```

Configure for GPG Keys

Stanza for each user in /etc/fwknop/access.conf:

```
SOURCE                      ANY
GPG_REMOTE_ID               7234ABCD
GPG_DECRYPT_ID              EBCD1234
GPG_ALLOW_NO_PW             Y
REQUIRE_SOURCE_ADDRESS      Y
REQUIRE_USERNAME            alice
FW_ACCESS_TIMEOUT           30
HMAC_KEY_BASE64             STQ9m03h
```

fwnop Fin

This is very powerful technology.

If you can keep your Internet-accessible services from being accessible to most attackers, they'll never get a shot at them.

SSH

SSH has a SOCKS proxy built right in.

If I want to encrypt my traffic and get it off the con network, but I haven't yet set up a VPN, I might just use the SOCKS proxy built into SSH.

ssh –D 31337 –p 19851 jay@his_ssh_server

Slides

Jay's slides at:

www.InGuardians.com/DefConJay.pdf

Larry's WI-FI Village slides at:

www.InGuardians.com/DefConLarry.pdf