# Internal

# Proxy APT

```
sudo nano /etc/apt/apt.conf.d/apt-proxy.conf
Acquire {  HTTP::proxy "http://172.31.13.247:3128/";  HTTPS::proxy
"http://172.31.13.247:3128/";}
sudo apt update && sudo apt upgrade -y
```

# 1. Scope

```
- Determine what can be done in and out of office hours
- Are there any 3rd party devices?
- Any 3rd party providers hosting services?
        - AWS
        - Azure
        - Digital Ocean
        - etc
```

# 2. Mapping and Reconnaissance

- Determine what IPs are alive
  - `sudo nmap -Pn -sS -iL <scope> --open -vvv -oN scope.live --max-retries=0 --top-ports 300 | tee -a scope.live.tee`
    - Scan for live hosts
  - `cat scope.live | grep -E 'report\s' | cut -d$' ' -f 5 > live.hosts`
    - Create a list of live hosts
- Determine what ports are open on each hosts
  - `sudo nmap -Pn -sS -iL live.hosts --open -vvv -oG scope.port --max-retries=0 -p- | tee -a scope.ports.tee`
    - Find all open ports, output to greppable
  - `python3 NmapGreppableParser.py scope.ports`
- Version Scan Each Host and its Open Ports
  - `mkdir hosts && cd hosts`
  - `Use output from NmapGreppableParser.py`

```
┌──(pentester㉿physical-jumpbox-01)-[~/scanning/nmap/hosts]
└─$ sudo nmap -Pn -sVC 172.16.1.25 -p25,80,443,515,631,9100,54921,54922,54923 -T4 -oN .nmap && \
sudo nmap -Pn -sVC 172.16.1.60 -p135,139,445,5040,5357,5985,7680,13001,13002,47001,49447,49664,49665,49666,4966
7,49669,49670,50478 -T4 -oN .nmap && \
sudo nmap -Pn -sVC 172.16.1.135 -p135,139,445,5040,5357,5700,5985,7680,47001,49664,49665,49666,49667,49668,4966
9,59368,59381 -T4 -oN .nmap && \
sudo nmap -Pn -sVC 172.16.1.143 -p135,139,445,5040,5357,5985,7680,47001,49664,49665,49666,49667,49668,49669,497
08,49927 -T4 -oN .nmap && \
sudo nmap -Pn -sVC 172.11.1.254 -p80,8090 -T4 -oN my.meraki.net.nmap && \
sudo nmap -Pn -sVC 172.11.1.25 -p25,80,443,515,631,9100,54921,54922,54923 -T4 -oN 172-11-1-25.lightspeed.jcvlfl
.sbcglobal.net.nmap && \
sudo nmap -Pn -sVC 192.168.21.103 -p22,5900 -T4 -oN .nmap && \
sudo nmap -Pn -sVC 192.168.21.104 -p5900 -T4 -oN .nmap && \
sudo nmap -Pn -sVC 192.168.21.105 -p5900 -T4 -oN .nmap && \
sudo nmap -Pn -sVC 172.16.1.148 -p21,25,53,80,88,110,135,143,389,443,445,587,636,1080,1433,1883,2022,3128,3389,
5985,5986,6666,9090,9389,46407 -T4 -oN .nmap
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-08-19 13:54 EDT
```

- Enumerate Web Services

```
cd ~/scanning
git clone https://github.com/urbanadventurer/WhatWeb.git
cd WhatWeb
# Http scan
./whatweb -i ~/scope -U 'Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/125.0.0.0 Safari/537.3' --
follow-redirect=always --log-verbose=ww.http --url-prefix='http://' --no-
errors
# https scan
./whatweb -i ~/scope -U 'Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/125.0.0.0 Safari/537.3' --
follow-redirect=always --log-verbose=ww.https --url-prefix='https://' --no-
errors
# whatweb urls
cat ww.http| grep "report for" | cut -d$' ' -f 4 > ww.urls

# Gowitness
# install gowitness
mkdir -p ~/scanning/gowitness/screenshots
cd ~/scanning/gowitness
wget https://github.com/sensepost/gowitness/releases/download/2.5.1/gowitness-
2.5.1-linux-amd64
chmod +x gowitness-2.5.1-linux-amd64
mv gowitness-2.5.1-linux-amd64 gowitness
sudo cp gowitness /usr/local/bin/gowitness

# iterate over urls, save screenshots
for url in $(cat /home/pentester/scanning/WhatWeb/ww.urls); do gowitness
single -F --js "console.log('ondefend')" --screenshot-path ./screenshots --
user-agent 'Mozilla/5.0 (Windows Phone 10.0; Android 6.0.1; Microsoft; RM-
1152) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/52.0.2743.116 Mobile
Safari/537.36 Edge/15.15254' "$url"; done
```

```
# 161
mkdir -p /home/pentester/scanning/161 cd /home/pentester/scanning/161
onesixtyone -i /home/pentester/scope -o scope.161
cat scope.161 | awk -F' \\[' '{print $1}' > 161.hosts

# snmp walk
snmpwalk
```

## SMB No Signing

```
mkdir -p relay
cd relay
nxc smb ~/scope --gen-relay-list no-signing.txt | tee -a relay-list.tee
```

## Webdav

```
nxc smb ~/scope -u '<user>' -p '<password>' -d '<domain>.<tld>' -M webdav |
tee -a webdav.tee
```

## Webdav

# Automated

## Nmap Scanning

```
#!/usr/bin/bash

# Set scope file
scope_file=/home/pentester/scope # Replace with input file

# Step 1: Initial Nmap Scan
sudo nmap -Pn -sS -iL "$scope_file" --open -vvv -oN scope.live -g 53 --max-
retries=0 --top-ports 20 -f --data-length 36 | tee -a scope.live.tee

# Step 2: Extract Live Hosts
cat scope.live | grep -E 'report\s' | cut -d' ' -f 5 > live.hosts

# Step 3: Full Port Scan on Live Hosts
sudo nmap -Pn -sS -iL live.hosts --open -vvv -oG scope.ports -g 53 --max-
retries=0 -p- -f --data-length 36 | tee -a scope.ports.tee
```

```
# Step 4: Parse Open Ports and Generate Commands
python3 /home/pentester/tools/NmapGreppable.py ./scope.ports >
open.ports.hosts

# Step 5: Process Each Command
mkdir -p hosts && cd hosts
while IFS= read -r command; do
  echo "Executing: $command"
  eval "$command"
done < ../open.ports.hosts
```

# Passive

```
# install docker
sudo apt install -y docker.io

# download and run PCreds
git clone https://github.com/lgandx/PCredz
cd PCredz

# alter the respodner config
sudo set 'g/SMB'
screen -S responder -d -m bash -c 'sudo responder -I eth0 -A'

# run ntlmrelayx to generate a computer account
screen -S ntlmrelayx -d -m bash -c 'sudo responder -I eth0 -A'

# kill old session and run ntlmrelayx to create a socks conn
screen -S ntlmrelayx -X quit
screen -S ntlmrelayx -d -m bash -c 'sudo ntlmrelayx -smb2support socks'
```

# SMB

```
nxc smb | tee -a smb.tee
```