CS 547: Foundation of Computer Security

S. Tripathy
IIT Patna

Previous *Class*

- Malicious code: Malware
  - Viruses
    - *Resident*
    - *Code*
      - *Spreading and payload*

Present Class

- Malicious code: Malware
  - Worms

- *Other malicious codes*
  - *Backdoor*
  - *Rootkit*
  - *Trojan horse and Logic Bomb*

- *Detection mechanisms*
  - *Signature based*
  - *Behaviour based*

Worms

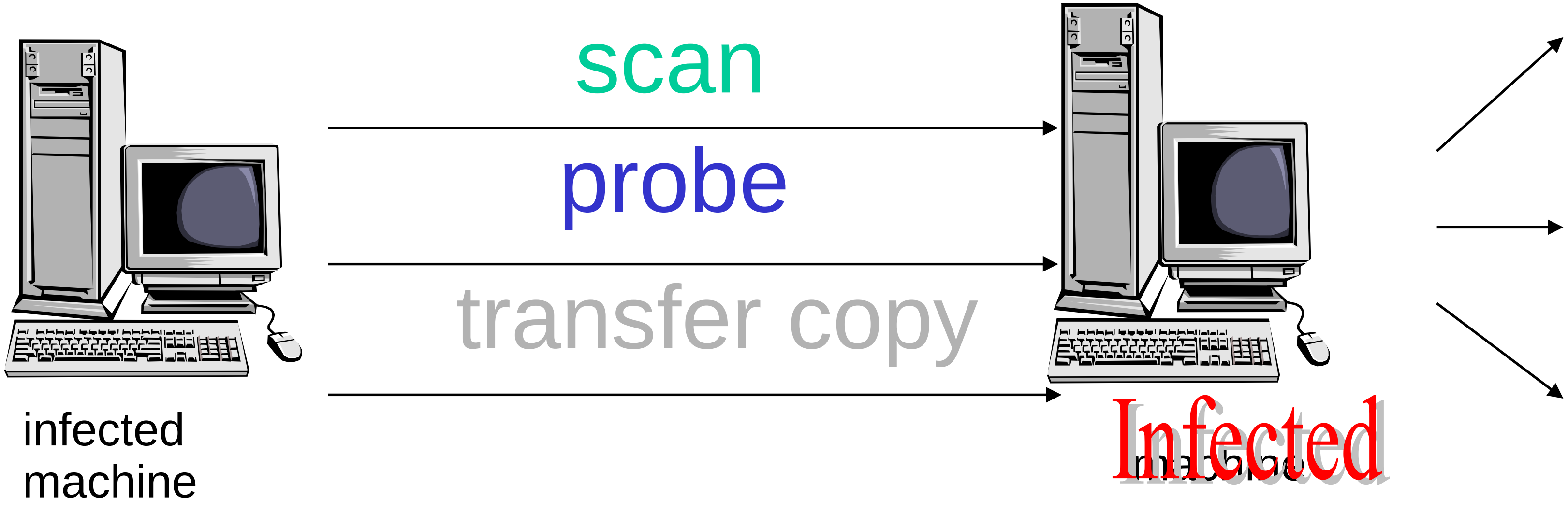
- A worm is a self-contained piece of code that can replicate with (little or) no user involvement
- Worms often use security flaws in widely deployed software as a path to infection
- Typically:
  - A worm exploits a security flaw in some software on your computer, infecting it
  - The worm immediately starts searching for other computers (on your local network, or on the Internet generally) to infect
  - There may or may not be a payload that activates at a certain time, or by another trigger

The Morris Worm of 1988

- First "worm" program :
  - Released by Robert T Morris of Cornell University
  - Affected DEC's VAX and Sun Microsystems's Sun 3 systems
- Spread
  - ~6000 victims i.e., 5-10% of hosts at that time
  - more machines disconnected from the net to avoid infection
- Cost
  - Some estimate: $98 million
  - Other reports: <$1 million
- Triggered the creation of CERT (Computer Emergency Response Team)

How an Active Worm Spreads

- Autonomous
- No need of human interaction



Hopping of Worm

- Worm program may hop from one machine to another by a variety of means:
  - By using the remote shell facilities, as provided by, ssh, rsh, rexec, etc., in Unix, to execute a command on the remote machine
    - By cracking the passwords and logging in as a regular user on a remote machine.
  - By using buffer overflow vulnerabilities in networking software.

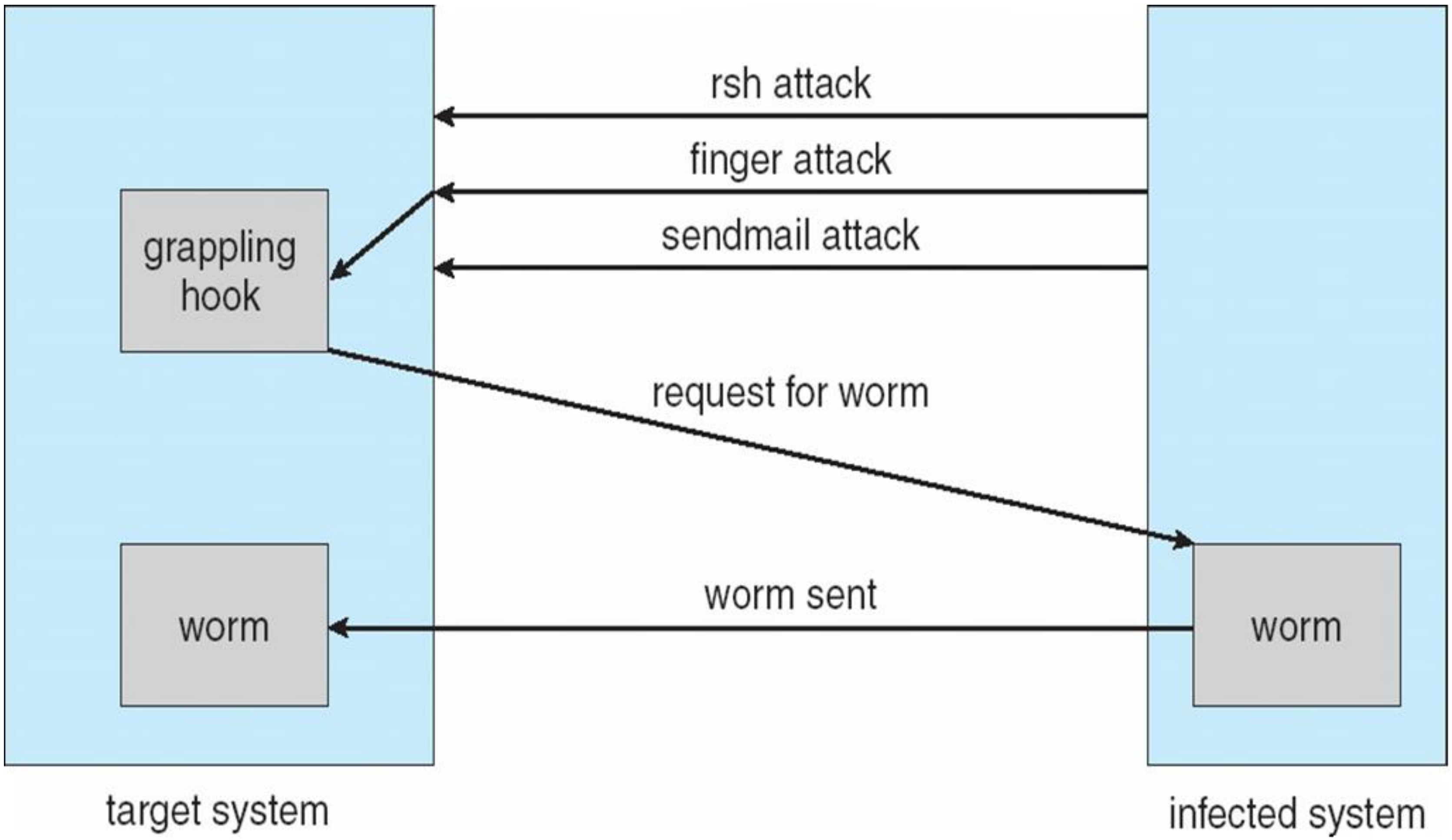
Internet Worm Description

- Two parts
  - Program to spread worm
    - look for other machines that could be infected
    - try to find ways of infiltrating these machines
  - Vector program (99 lines of C)
    - compiled and run on the infected machines
    - transferred main program to continue attack
- Security vulnerabilities
  - fingerd – Unix finger daemon
  - sendmail - mail distribution program
  - Trusted logins (.rhosts)
    - Weak passwords

Three ways the worm spreads

- Three ways the worm spreads by using
  - the remote shell facilities
  - Cracking the passwords
  - Buffer overflow vulnerabilities in networking software
    - Fingerd
      - Exploit a buffer overflow in the gets function
      - Apparently, this was the most successful attack
    - Sendmail
      - Exploit debug option in sendmail to allow shell access
    - Rsh
      - Exploit trusted hosts
      - Password cracking
  -

The Morris Internet Worm



Detecting Morris Internet Worm

- Files
  - Strange files appeared in infected systems
  - Strange log messages for certain programs
- System load
  - Infection generates a number of processes
  - Systems were reinfected => number of processes grew and systems became overloaded
    - Apparently not intended by worm's creator

Thousands of systems were shut down

Backdoor

- Software that allows access to a computer system bypassing the normal authentication procedures. For example
  - A special username and password hard-coded into the login program

- Such backdoors may be inserted by viruses, worms, Trojan horses or spyware.
  - A service listening on a particular IP port for remote instructions (e.g., Back Orifice)

Trusting Trust backdoor

- How to create an undetectable backdoor:
  - Change the compiler so that, when compiling the login program, it adds the hard-coded username/password check to the login program.
    - Thus, the login program source code looks completely normal.
  - As an extra twist, change the compiler so that, when compiling the compiler, it adds the code to the login program.
    - Thus, even if the compiler is recompiled, the backdoor will still be inserted.
    - And none of the source code reveals the backdoor.

Rootkit

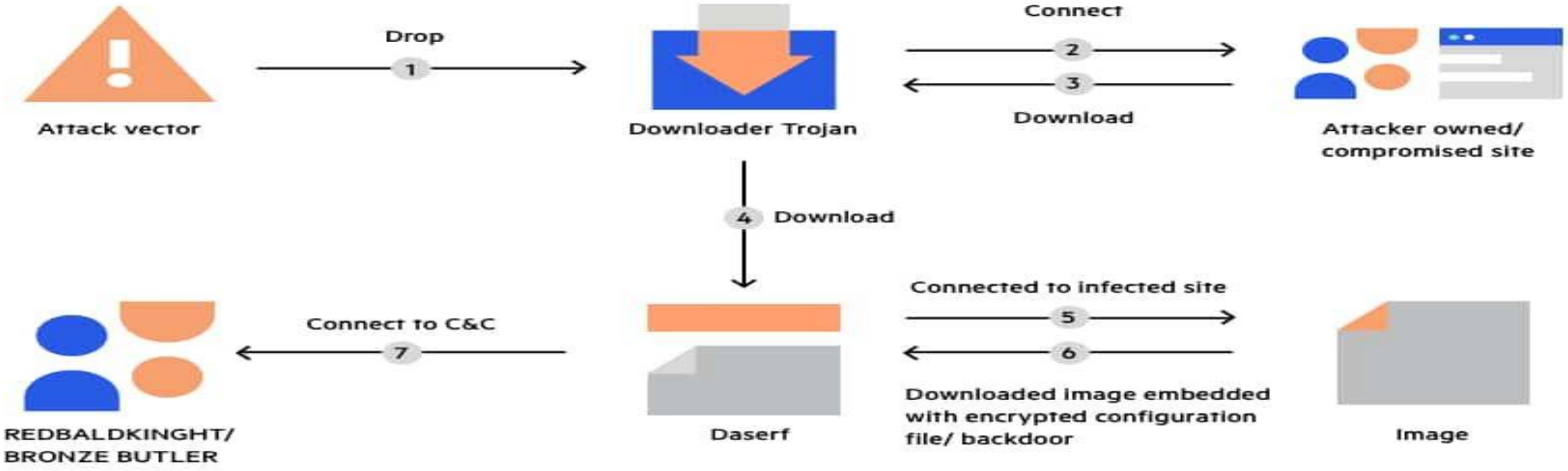
- After installing the backdoor, the cracker wishes to avoid being detected or removed by routine maintenance of the system. For that, she uses a rootkit.

- A rootkit is a set of modified versions of the usual utilities for administering the system, such as:
  - List all processes (unix: ps)
  - List logged-in users (unix: w, who)
  - List files (unix: ls)
  - Change passwords (unix: passwd)
  - Logging utilities

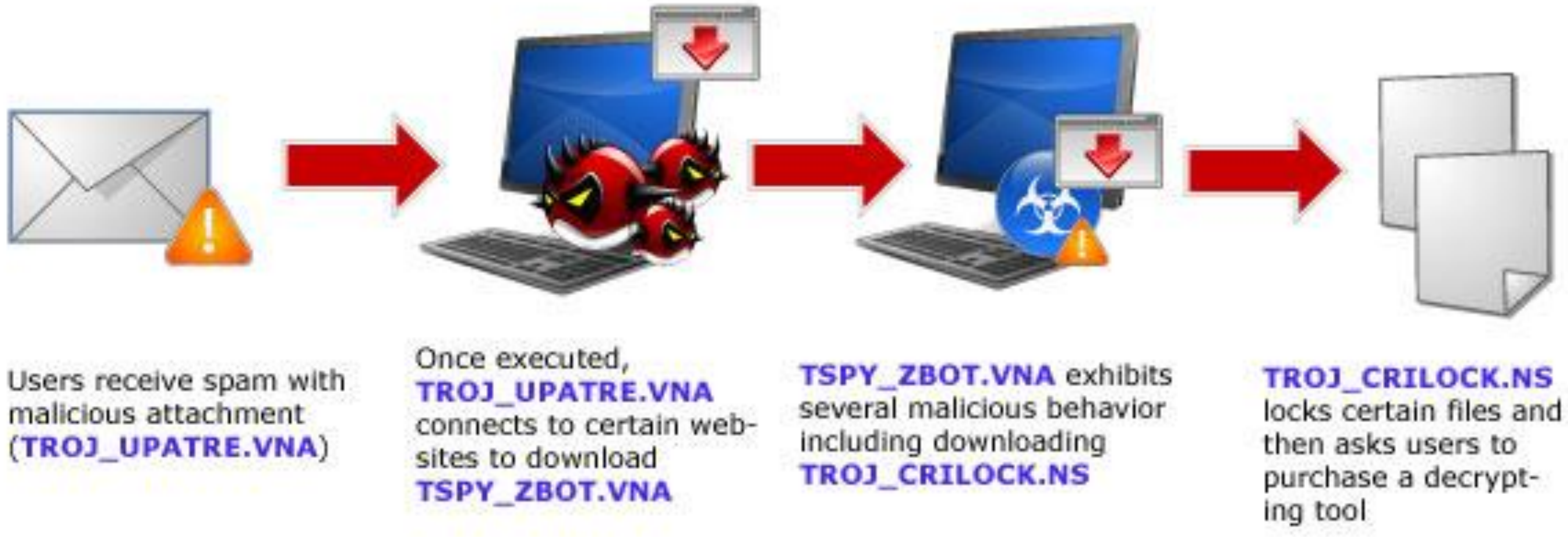
Trojan

– Undisclosed malicious functions that allow unauthorized access to the victim computer.

- Trojan Infection Methods :
  - A user is targeted by phishing or other types of social engineering, opens an infected email attachment or clicks a link to a malicious website
  - A user visits a legitimate website infected with malicious code
  - Attackers install a trojan by exploiting a software vulnerability, or through unauthorized access


Attack vector
Drop
1
Downloader Trojan
Connect
2
3
Download
Attacker owned/ compromised site
4 Download
Connected to infected site
5
6
Downloaded image embedded with encrypted configuration file/ backdoor
Image
Daserf
Connect to C&C
7
REDBALDKINGHT/ BRONZE BUTLER

Zeus/Zbot is a malware example



Users receive spam with malicious attachment (TROJ_UPATRE.VNA)

Once executed, TROJ_UPATRE.VNA connects to certain web-sites to download TSPY_ZBOT.VNA

TSPY_ZBOT.VNA exhibits several malicious behavior including downloading TROJ_CRILOCK.NS

TROJ_CRILOCK.NS locks certain files and then asks users to purchase a decrypt-ing tool

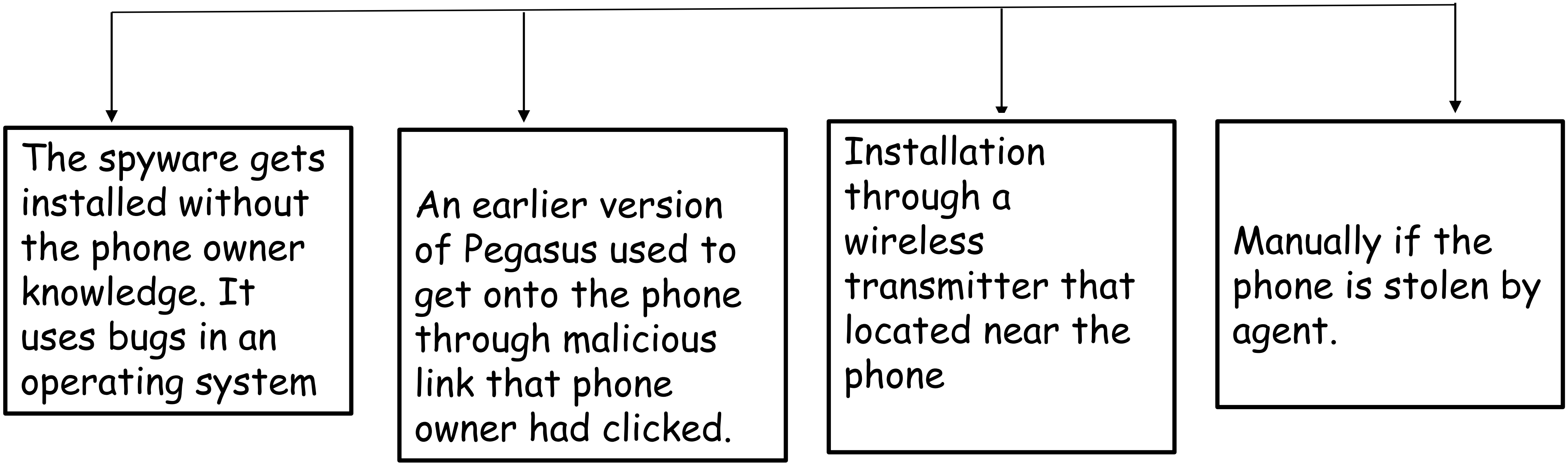
Spyware

- Malware that collects user information without their knowledge
  - Keyloggers: stealthly tracking and logging key strokes
  - Screen scrapers: stealthly reading data from a computer display
    - May also tracking browsing habit
    - May also re-direct browsing and display ads

Pegasus Spyware

- A Pegasus spyware, which affects Android and Ios operating system.
- can be installed without knowledge of the phone owner.
- then has access to the phone’s files, camera, and microphone, and it can also monitor the location.

How can Pegasus spyware infect a phone



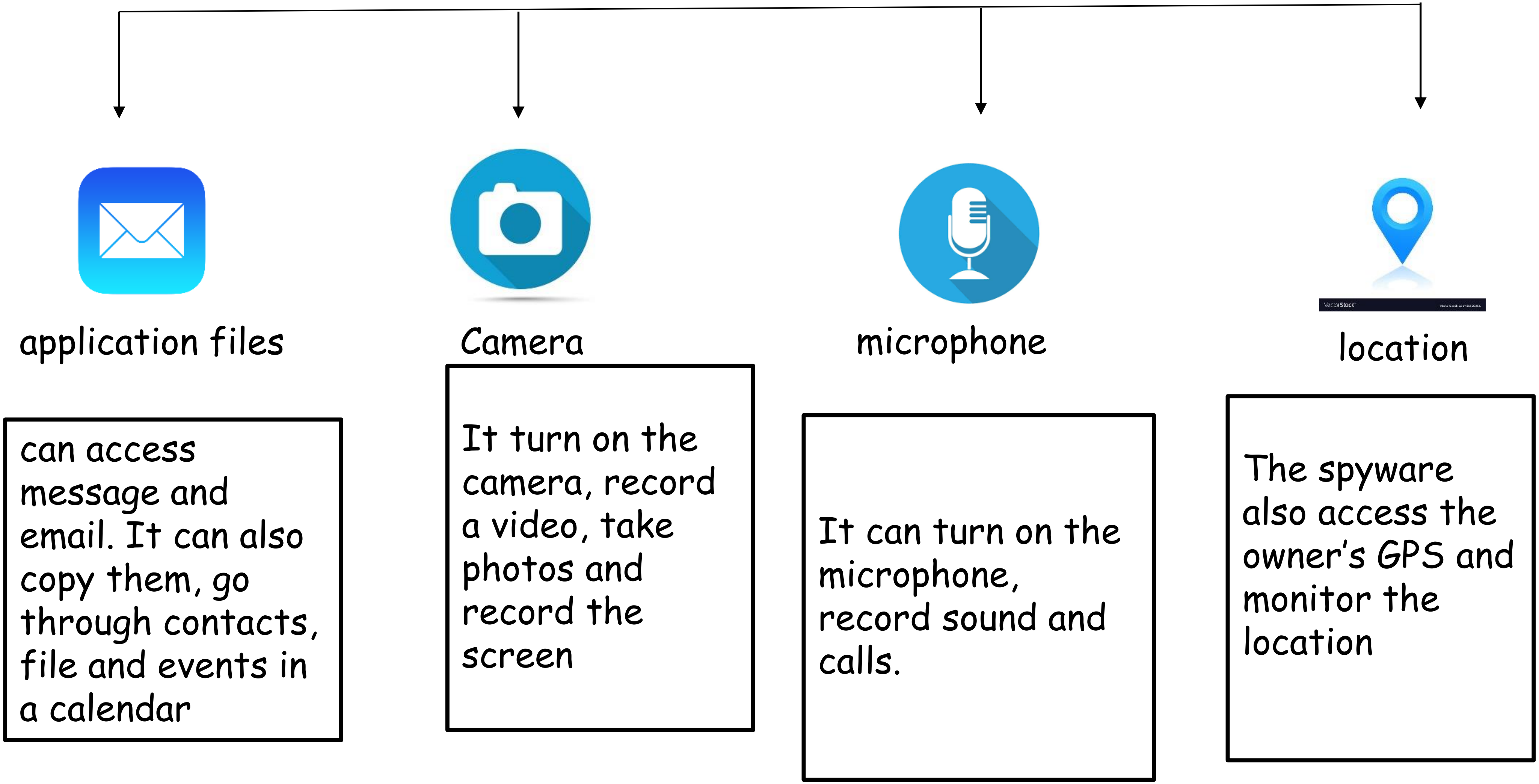
The spyware gets installed without the phone owner knowledge. It uses bugs in an operating system

An earlier version of Pegasus used to get onto the phone through malicious link that phone owner had clicked.

Installation through a wireless transmitter that located near the phone

Manually if the phone is stolen by agent.

What can Pegasus spyware access ?



application files

can access message and email. It can also copy them, go through contacts, file and events in a calendar

Camera

It turn on the camera, record a video, take photos and record the screen

microphone

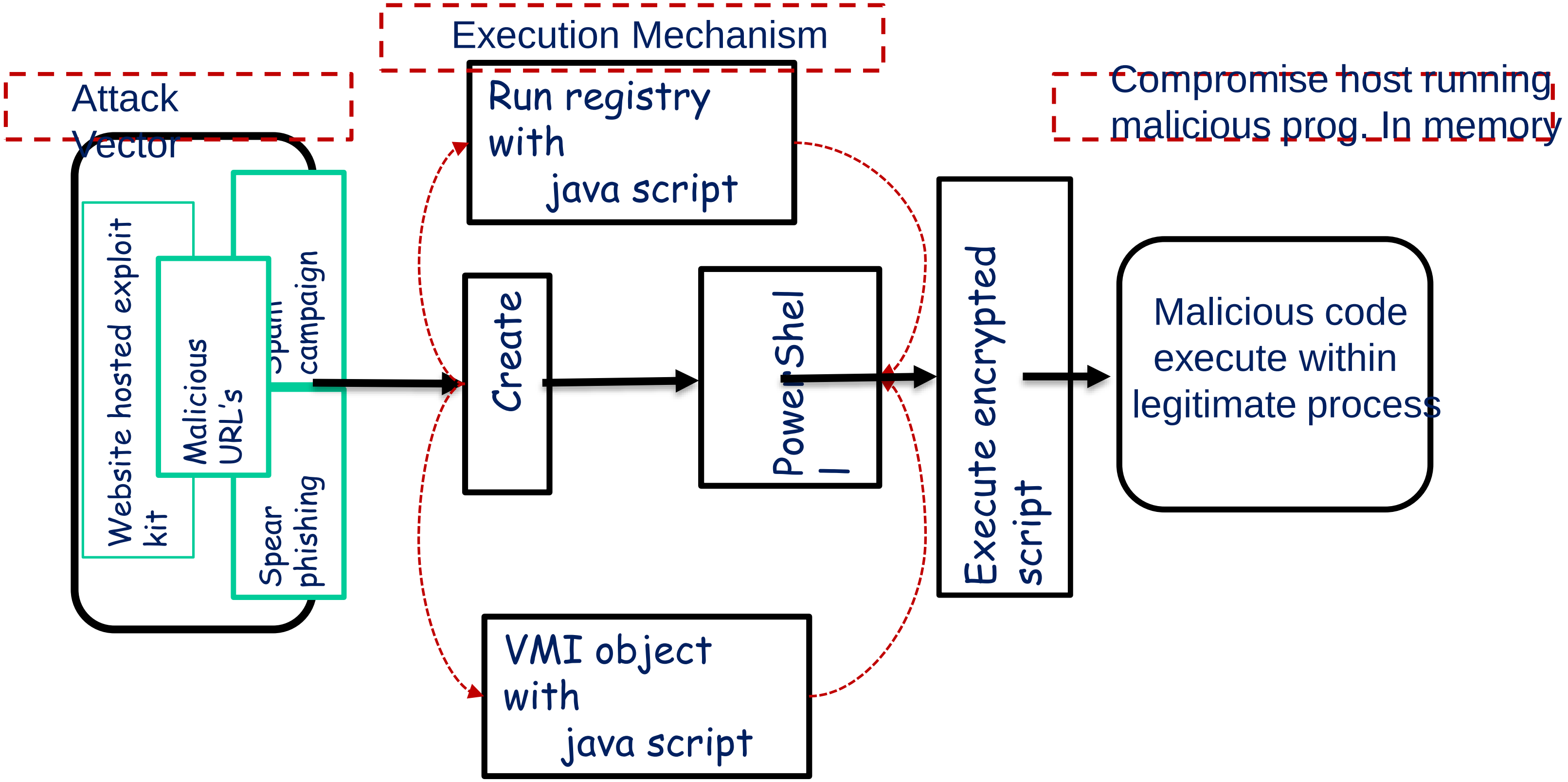
It can turn on the microphone, record sound and calls.

location

The spyware also access the owner's GPS and monitor the location

Fileless malware

- Execute malicious Java script/ VB Script directly in the memory evade the AV solutions.
- It can launch an attack across many phases of the attack life-cycle like reconnaissance, AV/VM detection, code execution


Execution Mechanism
Attack Vector
Compromise host running malicious prog. In memory
Website hosted exploit kit
Malicious URL's
Spam campaign
Spear phishing
Run registry with java script
Create
PowerShell
VMI object with java script
Execute encrypted script
Malicious code execute within legitimate process

Traditional malware & Fileless Malware

Techniques	Tradition file based malware	Fileless malware
Source Code	yes	No
malicious file	yes	No
malicious process	yes	No
Complexity	moderate	very high
detection complexity	moderate	very high
persistence	medium	low
file type	executable, script(pdf,word)	JS, VMI, Flash
Obfuscation method	Ecryp. File, Arch file, Exe file	Encoding, Unicode, whitespace, randomization
Target	Patch level combination	Path level combination
Antivirus detection	possible with known signature	Not possible
Sandboxes detection	Physical availability of file	Not possible
Behaviour, heuristic and machine learning	File based malware show abnormal behaviour in the system after compromise target host	Fileless attacks are designed to behave like benign process in the system

Anti-virus software

- Initially: signature detection.
- But signatures are not enough!
  - Pattern matching
  - Automatic learning
  - Environment emulation
  - Neural networks
  - Hidden Markov models

Generations of Anti-Virus Software

- *first generation*: simple scanners
  - requires a malware signature to identify the malware
  - limited to the detection of known malware
- *second generation*: heuristic scanners
  - uses heuristic rules to search for probable malware instances
  - another approach is integrity checking
- *third generation*: activity traps
  - memory-resident programs that identify malware by its actions rather than its structure in an infected program
- *fourth generation*: full-featured protection
  - packages consisting of a variety of anti-virus techniques used in conjunction include scanning, activity trap components and access control capability

Anti-virus software: TbScan

- TbScan looks at the following characteristics:
  - F = Suspicious file access. Might be able to infect a file.
  - R = Relocator. Program code will be relocated in a suspicious way.
  - A = Suspicious Memory Allocation. The program uses a non-standard way to search for, and/or allocate memory.
  - N = Wrong name extension. Extension conflicts with program structure.
  - S = Contains a routine to search for executable (.COM or .EXE) files.
  - # = Found an instruction decryption routine. This is common for viruses but also for some protected software.
  - E = Flexible Entry-point. The code seems to be designed to be linked on any location within an executable file. Common for viruses.
  - L = The program traps the loading of software. Might be a virus that intercepts program load to infect the software.
  - D = Disk write access. The program writes to disk without using DOS.
  - M = Memory resident code. This program is designed to stay in memory.
  - ! = Invalid opcode (non-8088 instructions) or out-of-range branch.
  - T = Incorrect timestamp. Some viruses use this to mark infected files.

TbScan (continued)

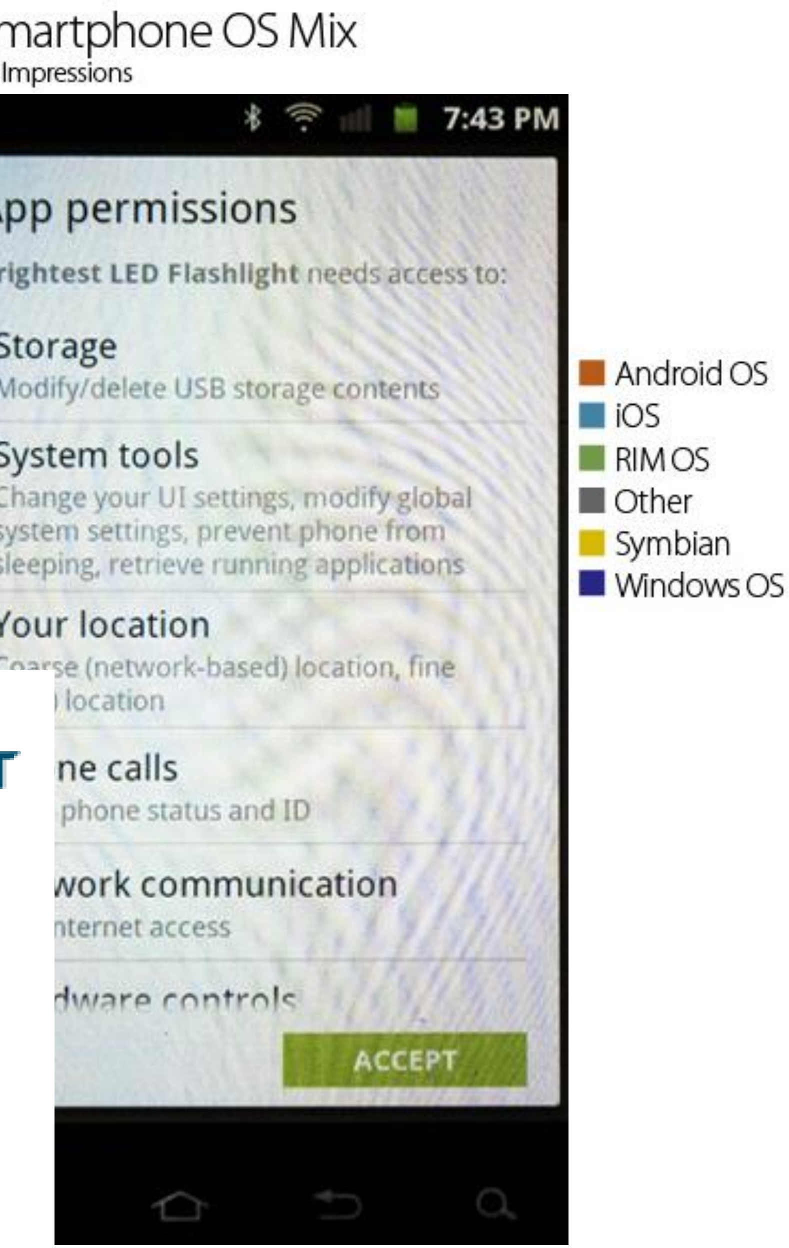
- TbScan (continued)
  - J = Suspicious jump construct. Entry point via chained or indirect jumps. This is unusual for normal software but common for viruses.
  - ? = Inconsistent exe-header. Might be a virus but can also be a bug.
  - G = Garbage instructions. Contains code that seems to have no purpose other than encryption or avoiding recognition by virus scanners.
  - U = Undocumented interrupt/DOS call. The program might be just tricky but can also be a virus using a non-standard way to detect itself.
  - Z = EXE/COM determination. The program tries to check whether a file is a COM or EXE file. Viruses need to do this to infect a program.
  - O = Found code that can be used to overwrite/move a program in memory.
  - B = Back to entry point. Contains code to re-start the program after modifications at the entry-point are made. Very usual for viruses.
  - K = Unusual stack. The program has a suspicious stack or an odd stack.

Android malware

- Target regular users (non-rooted)
- Usual uses:
  - Steal personal data including, not limited to
    - Contacts
    - Banking details
    - Secrets (files)
  - Mine crypto-currency
  - Use for DDoS botnets
  - Ransom (blackmail)
  - Destroy device

Smartphone OS Mix

By Impressions



App permissions

Brightest LED Flashlight needs access to:

Storage

Modify/delete USB storage contents

System tools

Change your UI settings, modify global system settings, prevent phone from sleeping, retrieve running applications

Your location

Coarse (network-based) location, fine) location

ne calls

phone status and ID

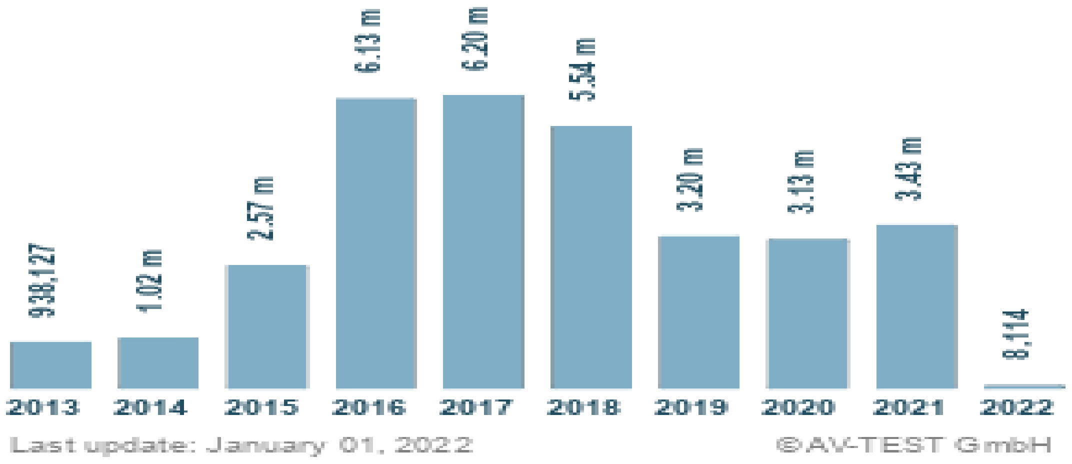
work communication

nternet access

dware controls

Development of Android malware

AV-TEST



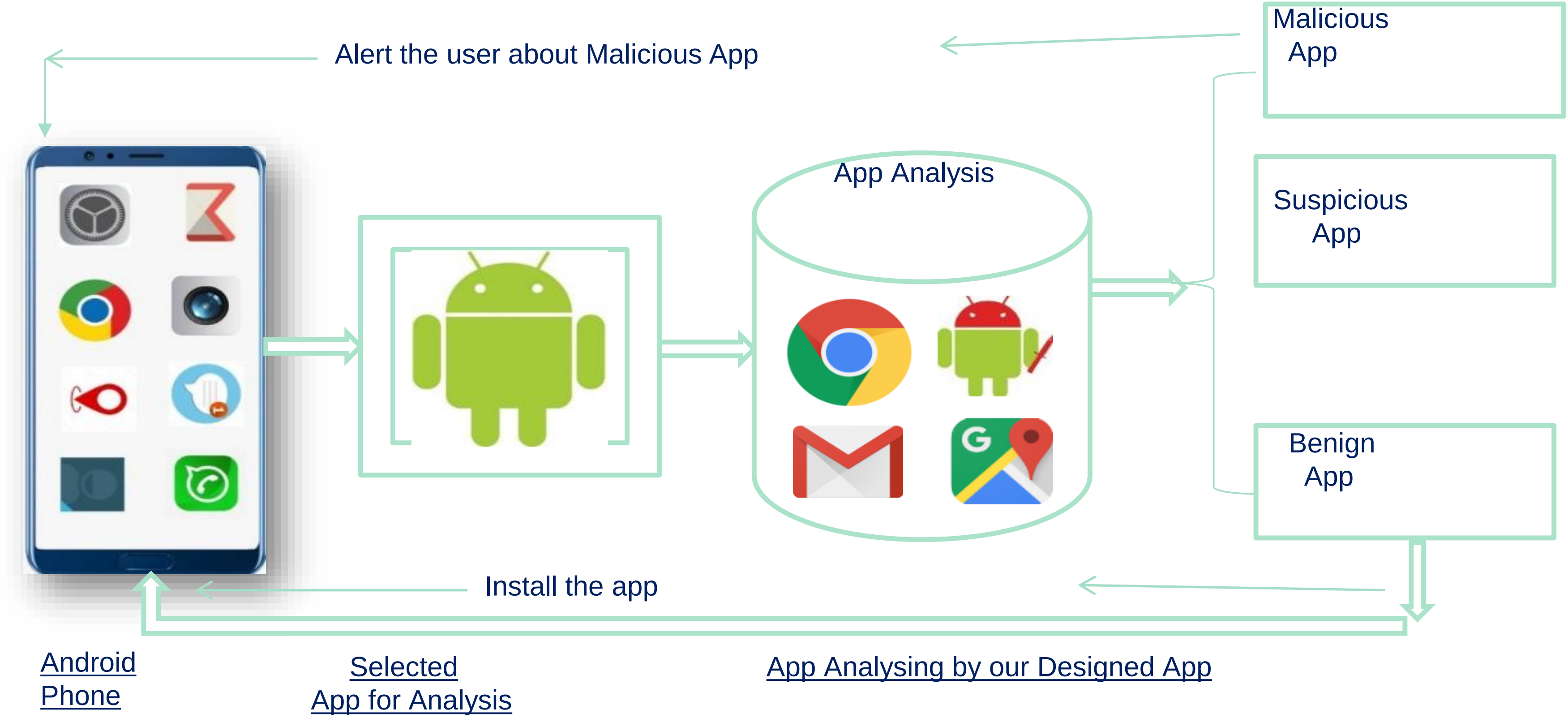
Last update: January 01, 2022

© AV-TEST GmbH

Current Android Malware

- **Description**
- **AccuTrack**
  This application turns an Android smartphone into a GPS tracker.
- **Ackposts**
  This Trojan steals contact information from the compromised device and uploads them to a remote server.
- **Acnetdoor**
  This Trojan opens a backdoor on the infected device and sends the IP address to a remote server.
- **Adsms**
  This is a Trojan which is allowed to send SMS messages. The distribution channel ... is through a SMS message containing the download link.
- **Airpush/StopSMS**
  Airpush is a very aggresive Ad-Network.
- …
- **BankBot**
  This malware tries to steal users' confidential information and money from bank and mobile accounts associated with infected devices.

ADAM: Automatic Detection of Android Malwar


Alert the user about Malicious App
Malicious App
App Analysis
Suspicious App
Benign App
Install the app
Android Phone
Selected App for Analysis
App Analysing by our Designed App

Somanath Tripathy, Narendra Singh, and Divyanshu Singh, 14th International Conference on Security for Information Technology and Communications – SECITC 2021

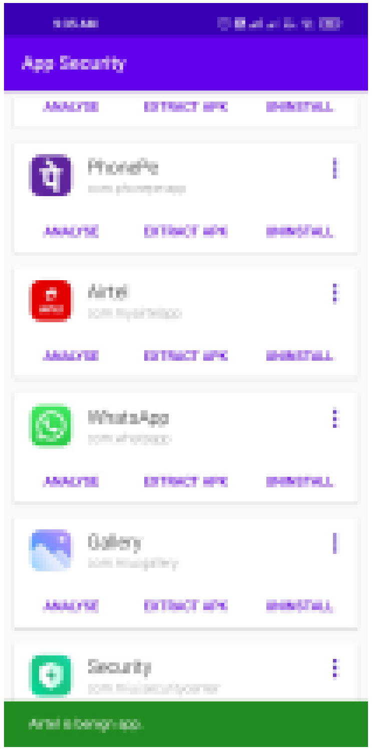
Deployment

DL Model is deployed in Smart Device using Tensor flow Lite

TensorFlow Lite is TensorFlow's lightweight solution for mobile and embedded devices. It enable on-device machine learning inference with low latency and small binary size

Procedure to deploy DL model

- Pick our pre-trained model
- Convert the model into TensorFlow Lite format
- Run our model on the device with the TF Lite interpreter
- Optimized the model using Model Optimization Toolkit



PhonePe

Airtel

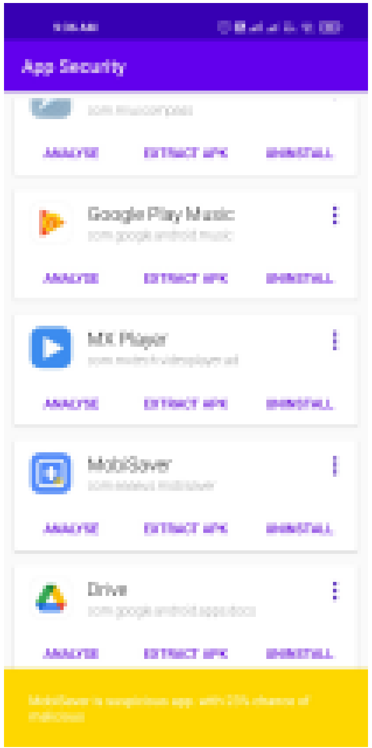
WhatsApp

Gallery

Security

Airtel is benign app.

(a) benign



Google Play Music

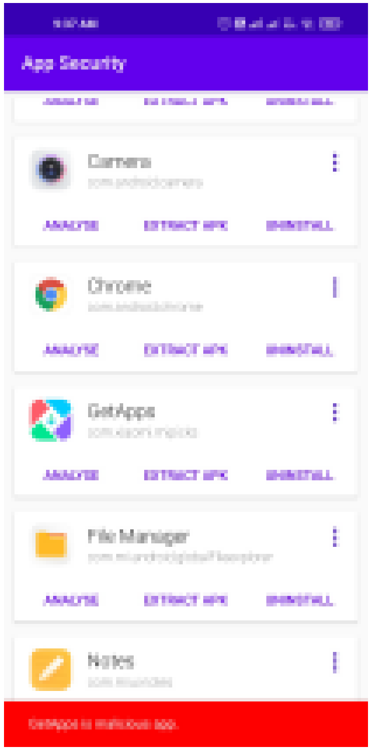
MX Player

MobiSaver

Drive

MobiSaver is suspicious app. with [illegible] chance of malicious

(b) suspicious



Camera

Chrome

GetApps

File Manager

Notes

GetApps is malicious app.

(c) malign

Thanks