Topics in Mobile Security

Ankit Pradhan

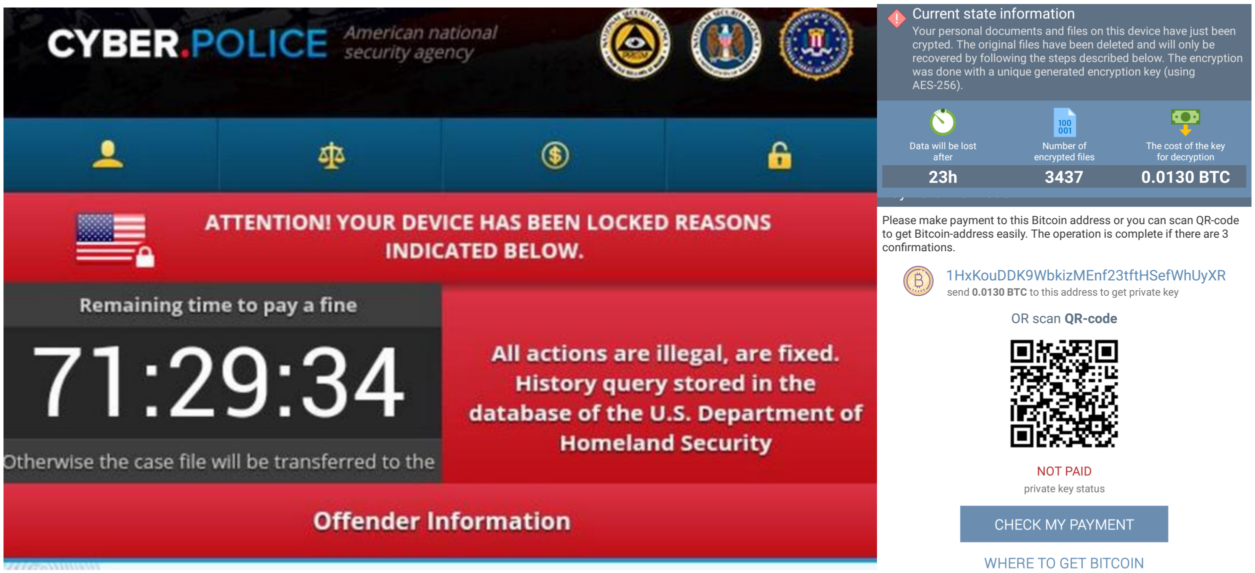
School of Electrical Sciences
Indian Institute of Technology Bhubaneswar
ap36@iitbbs.ac.in

January 23, 2019

Contents

1 Ransomware on Mobile Devices

1.1 Success of Ransomware



CYBER.POLICE American national security agency

ATTENTION! YOUR DEVICE HAS BEEN LOCKED REASONS INDICATED BELOW.

Remaining time to pay a fine

71:29:34

Otherwise the case file will be transferred to the

All actions are illegal, are fixed. History query stored in the database of the U.S. Department of Homeland Security

Offender Information

Current state information

Your personal documents and files on this device have just been crypted. The original files have been deleted and will only be recovered by following the steps described below. The encryption was done with a unique generated encryption key (using AES-256).

Data will be lost after	Number of encrypted files	The cost of the key for decryption
23h	3437	0.0130 BTC

Please make payment to this Bitcoin address or you can scan QR-code to get Bitcoin-address easily. The operation is complete if there are 3 confirmations.

1HxKouDDK9WbkizMEnf23tftHSefWhUyXR

send **0.0130 BTC** to this address to get private key

OR scan **QR-code**

NOT PAID

private key status

(a) Dogspectus

(b) DoubleLocker

Figure 1: Screenshot of the ransom note displayed on an infected device by Android ransomwares

Since its inception in 1996 [1], ransomware has evolved into a more terrifying threat than what its founders Adam Young and Moti Yung would have predicted. Due to the exponential growth in the use of smart phones in recent years and imminent monopoly of Android with a market share of 82.8% [2], as an open source and customizable operating system platform, has created an attractive whirlpool for attackers to disseminate ransomware.

Ransomware is a type of malware that threatens to publish the victim's data or perpetually block access to the hijacked resource unless a ransom is paid. On Android, there are three general categories of ransomware:

- **Lock-screen ransomware** - display an image covering the screen and block any access to the device until the ransom is paid.
- **PIN lockers** - misuse the built-in operating system protective mechanism and modify the existing PIN for unlocking the device without the knowledge of the user.
- **Crypto-ransomware** - user's data gets encrypted and the user has to pay the ransom within a stipulated time to obtain the secret key to decrypt the data.

Ransomware has capabilities to wipe the affected device, open an arbitrary URL in the phone's browser, send an SMS message to any or all contacts, steal received SMS messages, steal contacts, display a different ransom message, enable or disable mobile data and Wi-Fi at will or even track the user's GPS location. Some prominent ransomware which surfaced on Android devices include DoubleLocker (Figure 1b), Charger, Jisut, Dogspectus (Figure 1a), Lockerpin, Simplocker, etc [3]. With sharp increase in number of ransomware attacks in 2017 on Desktops and Laptops by WannaCry, Petya and Bad Rabbit, mobile ransomware also showed their impact on Android devices. All these events saw global recognition with 2017 being named as the year of ransomware.

1.2 Future of Ransomware

While 2017 has been the year of ransomware, future might be a witness to more widespread, critical and dangerous attacks. A careful analysis of arising threats will shed insight to alternate directions in which ransomware can be used by criminals. One such direction includes use of ransomware as a diversion enabling background device infiltration, data scraping and compromise and even illegal transfer of funds. Ransomware can act as a medium for blackmail via exposure of victim's private information which damage reputation or reduce opportunities in the social and professional life of the victim. On enterprise infrastructures, ransomware can virtually target both small and large organizations triggering huge financial losses in the form of ransom. Recent developments indicate that future ransomware will have more targeted, sophisticated and atomic spread mechanisms through phishing and spam emails due to their demonstrated success leading unsuspecting users to malicious websites and attachments [4].

1.3 Impact of Ransomware on the Internet of Things (IoT)

Initially IoT ransomware were not given sufficient attention since they differed from traditional ransomware in the following two ways [6]:

- **Irreversibility of classic ransomware** wherein if the files in a PC, laptop or smartphone becomes inflicted with ransomware, the only possible solution rested in paying the ransom and recovering the secret key used to encrypt the data. IoT data on the other hand is stored in the cloud and the device itself has little value. Hence IoT ransomware attackers resort to Locker variants which lock the device and demand ransom for unlocking. Resetting of the device provides a simple solution to this problem.

- **Perspective of the attackers** varies with respect to differences in IoT devices which make it challenging for hackers to target effectively. Also, headless nature of IoT devices (lacking user interface) makes it difficult for attackers to display the ransom note and thus they have to find alternative ways of discovering user's email or hack into the IoT device controlling app.

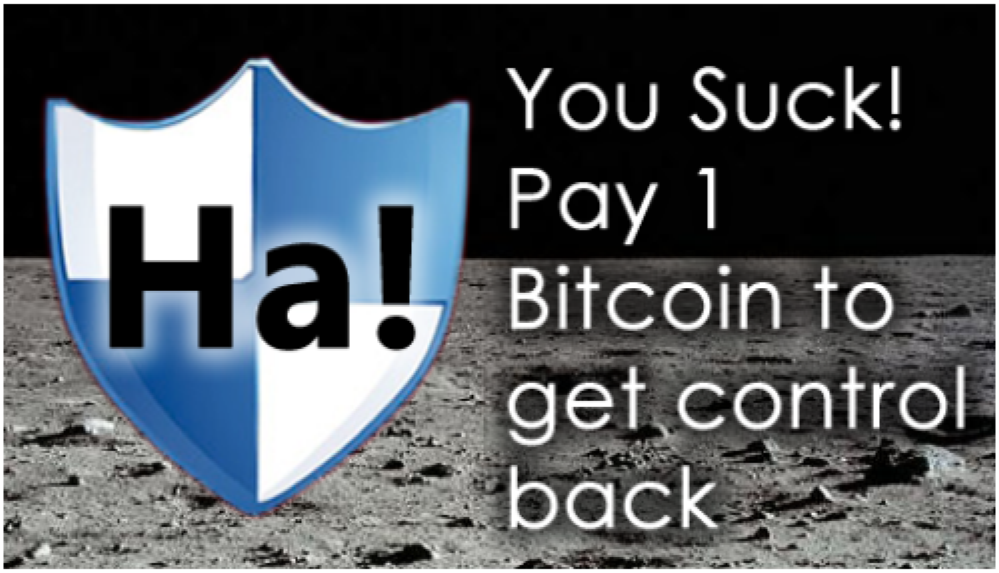
But the threat of IoT ransomware lies in the attack's timing and level of criticality. The proof-of-concept ransomware attack on Smart-Thermostat in 2016 by Andrew Tierney and Ken Munro demonstrated that critical timing in locking the thermostat (i.e., when user is away from home and cannot return immediately) can force the user to pay the ransom since otherwise the thermostat can be set at high temperature which can lead to fire accidents or huge electricity bills. Figure 2 shows the message displayed by the ransomware on Smart-Thermostat demanding 1 Bitcoin ransom. Other devices which are vulnerable to ransomware include Smart-TVs, CCTV cameras, Ticket Vending machines, Industrial Plants, Smart Grids, Autonomous Vehicles, Medical Devices (Pacemakers, Defibrillators), etc, [5].

1.4 Protective measures against Android Ransomware

Awareness of ransomware threats is the first step towards defense against future ransomware. ESET's technical report on Android Ransomware [3] lists a few preventive measures like:

- Avoiding unofficial app stores to download apps

- Using a trusted mobile security app installed and kept up-to-date



You Suck!
Pay 1
Bitcoin to
get control
back

Figure 2: Ransomware message displayed on Smart-Thermostat

- Keeping backup of all important data from the device (special counter against Crypto-ransomware)

But if ransomware has managed to infect the device, the following actions can be taken:

- Booting the device into Safe Mode for most simple lock-screen ransomware families and uninstalling the malicious application later.
- Resetting the lock using Google's Android Device Manager or an alternate Mobile Device Management (MDM) solution in case if the ransomware locked the device using PIN or password screen lock functionality by obtaining device administrator rights. A factory reset, deleting all data on the device, can be used as the last resort.
- Contacting security provider's technical support for help in case of Crypto-ransomware attack since depending on the ransomware variant, decryption of files may not be even possible.

The report strongly advises not to pay the ransom since they have seen many ransomware variants where code for decrypting files or uninstalling the lock-screen was missing. Thus paying the ransom does not yield a solution.

2 Biometric Sensors

One of the primary uses of biometric sensors on mobile devices is aimed at user/owner authentication even though recent sensors have capabilities to monitor pulse rate [7], blood pressure and remote authentication [9]. Provision of non-repudiable authentication has presented biometric sensors as a trusted solution in health-care, manufacturing, retail, finance, military, and education. In this light, Fingerprint sensors have become an integral component of smartphones with security enabled by faster and accurate readings. Due to the distinctiveness of fingerprints from person to person and non-intrusive nature of fingerprint scanners, the demand for PIN, pattern and gesture lock mechanisms in smartphones have reduced.

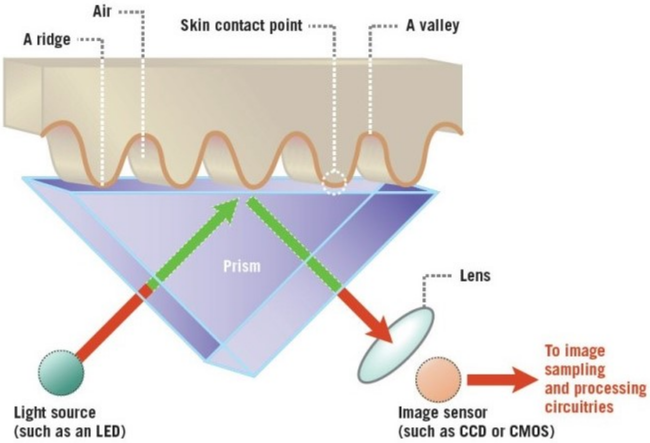

Air
A ridge
Skin contact point
A valley
Prism
Lens
To image sampling and processing circuitries
Light source (such as an LED)
Image sensor (such as CCD or CMOS)

Figure 3: Schematic diagram of an Optical sensor

Fingerprint sensors were initially designed in the form of **Optical Scanners** where an optical image of the fingerprint is taken and compared using pattern-matching algorithms with the existing/registered fingerprints of the user as shown in Figure 3. With higher sensor resolution, finer details of ridges and valleys on the surface of one's finger can be captured by multiple LEDs lighting the scanned region thus increasing the level of security [8]. But many major drawbacks including use of high resolution 2D pictures and prosthetics have reduced people's trust on its security and they are slowly been replaced by **Capacitive Scanners**.

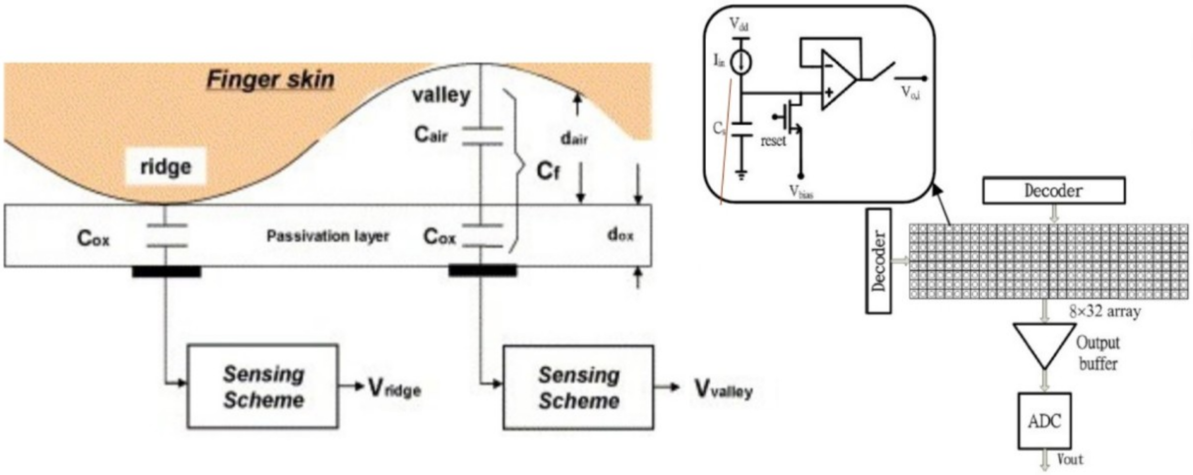


Figure 4: Architecture of capacitive fingerprint scanning chip

With arrays of capacitor circuits storing electrical charge, the capacitive scanners record slight charge variations due to ridges on the fingers and these variations are amplified and recorded by digital circuitry for analysis of distinct and unique attributes as shown in Figure 4 which we shall refer as *Electrical Signature*. These signatures are difficult to replicate using prosthetics as changes in charge of capacitors vary with material used in the prosthetic. But due to the high incurring costs on integrating large number of components in detection circuit, some implementations use partial prints in matching with the registered fingerprint while others allow multiple trials with different fingers. These may seem secure as every fingerprint is unique, but there are high odds (1 in 50,000) for small sections of different fingerprints to match [12] as shown by research in N.Y.U. Tandon School of Engineering. But due to its inherent security, capacitive scanners are used in most current day mobile devices including Galaxy S8, HTC U11, LG G6, and others [8].

Recent advances in biometric sensors include Qualcomm Snapdragon Sense ID 3D Fingerprint technology which employs Ultrasonic scanners to capture 3D details of surface of the finger than capacitive scanners [13].

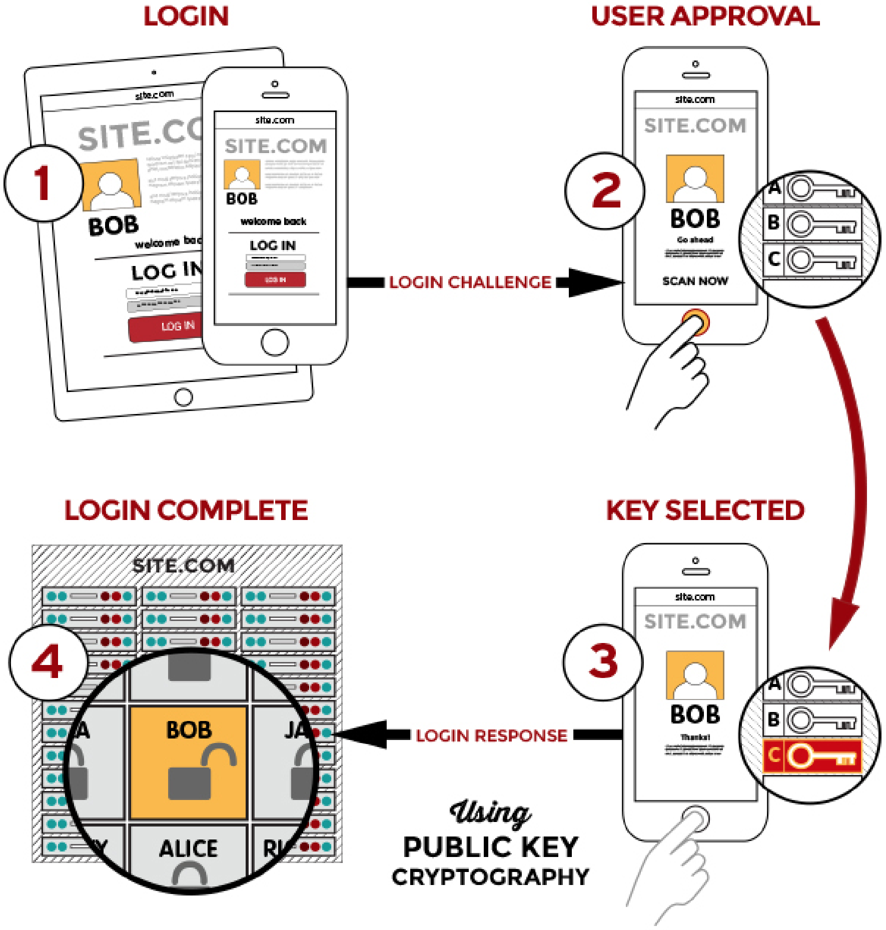


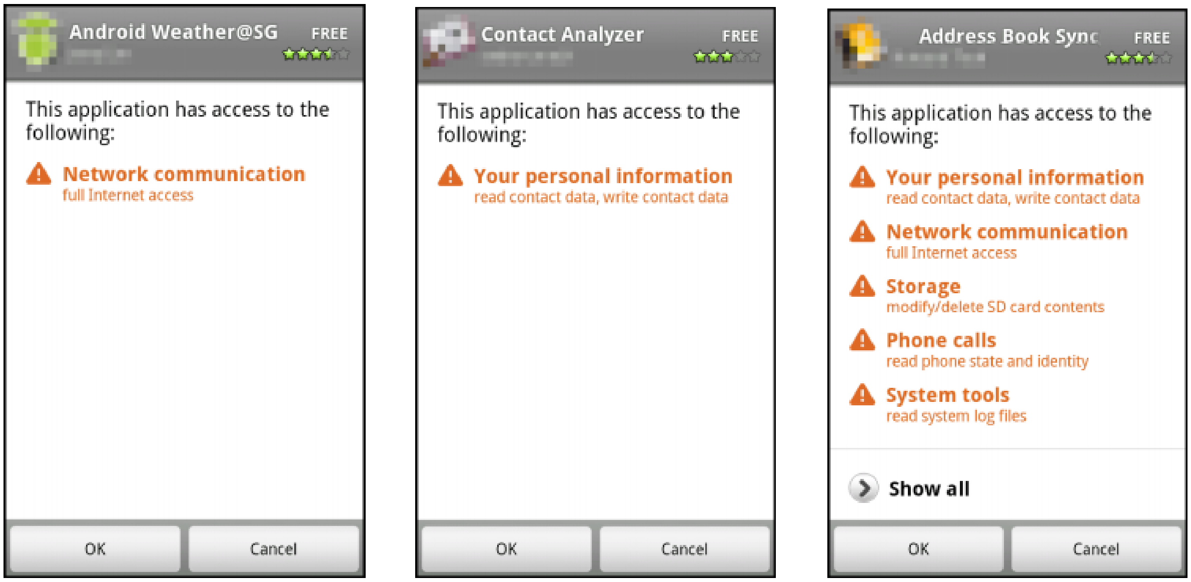
Figure 5: Operation of FIDO login process without sending any personally identifiable information.

Detection algorithms compares ridge splits, ridge line end locations and other minutiae to reduce processing power required for detection and avoid errors due to smudging. This information is kept secure in ARM processors using its Trusted Execution Environment (TEE) based TrustZone technology disabling apps operating in the regular operating system environment from accessing it. Strong cryptographic protocols have been designed by the **FIDO** (**F**ast **ID**entity **O**nline) **Alliance** to enable password-less authentication between hardware devices. This allows users to log into a website or online shop using fingerprint without sharing unique fingerprint data as depicted by Figure 5.

Current research in vulnerabilities in biometric sensors has been progressing steadily with works such as [10]. With the advent of deep learning works like [11] pose serious questions to technological advancement of fingerprint sensors and bio-inspired authentication systems. Nevertheless improving technology and innovative and secure solutions to mobile authentication will enable biometric sensors to play a crucial role in ensuring security in the future.

3 User Permission

Permission based mechanisms restricting accesses of third-party applications to critical resources on a mobile device form the backbone for mobile security. Due to the large market share enjoyed by Android, permission based security has been researched in detail on Android platform and it has been widely criticized for its coarse-grained permission control and difficulty in permission management by developers, marketers, and end-users [15].



This application has access to the following:

Network communication
full Internet access

This application has access to the following:

Your personal information
read contact data, write contact data

This application has access to the following:

Your personal information
read contact data, write contact data

Network communication
full Internet access

Storage
modify/delete SD card contents

Phone calls
read phone state and identity

System tools
read system log files

(a) Network access requirement (b) Contact access requirement (c) Possibly suspicious Application

Figure 6: The figure shows the permissions of three example applications taken from the Android Market. Applications declare their required permissions and wait for user decision on installing the application [14].

Android restricts accesses to critical resources using permissions which is a unique text string defined by Android or third party developers. According to the documentation for Android developers, there are currently 130 permissions, which are defined in Android operating system, ranging from dialing a phone number (CALL_PHONE), access to camera (CAMERA), full access to the Internet (INTERNET), and even disabling the phone function permanently (BRICK) [15].

Figure 6 shows screen-shots of applications requiring user permission for varying protection levels. Generally, the following four protection levels can be associated with a permission [15]:

- **Normal:** A low-risk permission allowing applications to access API calls with no harm to users (e.g., SET_WALLPAPER).
- **Dangerous:** A high-risk permission allowing applications to access potential harmful API calls which can leak private user data or control the device (e.g., READ_CONTACTS).
- **Signature:** A permission which is granted if its requesting application is signed with the same certificate as the application defining the permission.

- **Signature-or-system:** A permission which is granted only if its requesting application is in the same Android system image or is signed with the same certificate as the application defining the permission.

Major issue with most applications resides in the fact that at install-time, the user must grant all permissions requested by the application together or deny them together and once the application is installed and permissions approved by the user, it owns the permission throughout its lifetime and does not need to request it again at run-time.

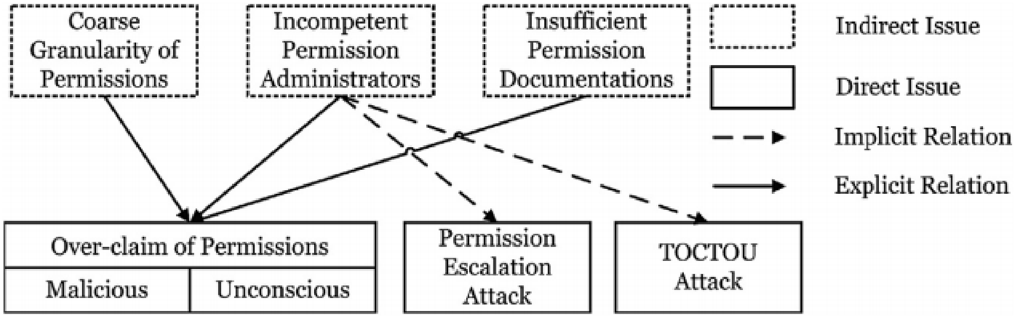

Coarse Granularity of Permissions
Incompetent Permission Administrators
Insufficient Permission Documentations
Indirect Issue
Direct Issue
Implicit Relation
Explicit Relation
Over-claim of Permissions
Malicious
Unconscious
Permission Escalation Attack
TOCTOU Attack

Figure 7: Relationship among issues in Android permission mechanism

The following are the main issues of permission based android applications [15], the relationship among which is depicted in Figure 7:

- **Coarse granularity of permissions** - permission give an application arbitrary accesses to certain resources. (e.g., INTERNET permission allows an application to send HTTP(S) requests to all domains, and connect to arbitrary destinations and ports.

- **Incompetent permission administrators** - developer may not know in detail what is at risk for end-users if the application is granted with required permissions.

- **Insufficient permission documentation** - even though Google Inc. provides sufficient documentation for Android application developers, the content on usage of permissions on Android platform is limited.

- **Over-claim of permissions** - serious problems such as potential privacy leakage and financial losses can result breaking the principle of least privilege (PLP).

- **Permission escalation attack** - allows a malicious application to collaborate with other applications so as to access critical resources without explicitly requesting for corresponding permissions. Figure 8 gives an overview of permission escalation attack. Marforio et al. [14] show situations where attacks by colluding applications can occur through communication over overt and covert communication channels.

- **TOCTOU attack** - naming collision can enable the TOCTOU (Time of Check to Time of Use) attack. Since, permissions in Android are represented as strings, and any two permissions with the same name string are treated equivalent even if they belong to different applications, a malicious developer can exploit this flaw by injecting permission with same name as required permission.

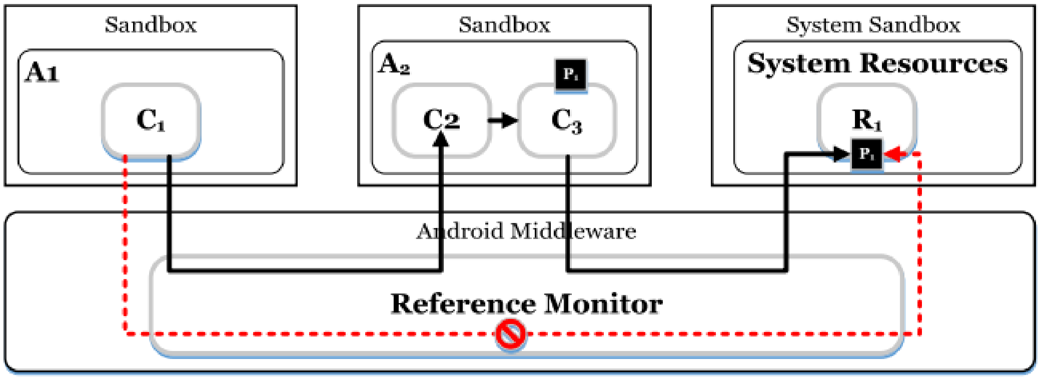


Figure 8: Permission Escalation Attack

Currently, many security extensions to Android's middleware exists like Kirin, Saint, TaintDroid, and QUIRE to mitigate permission based attacks. **XManDroid (eXtended Monitoring on Android)**, introduced by Bugiel et al. [17], is a security framework that extends the monitoring mechanism of Android to detect and prevent application-level privilege escalation attacks at run-time. Jiang et al. [16] have recently introduced multi-layered permission-based security extension scheme on Android platforms. They propose a usage and access control model and using ARM TrustZone security extension mechanism prevent permission leakage by providing a mandatory access control on Android middleware, Linux kernel, and hardware layers. With better fine-grained access control over permissions requested by applications, many mobile security problems can be addressed in the future.

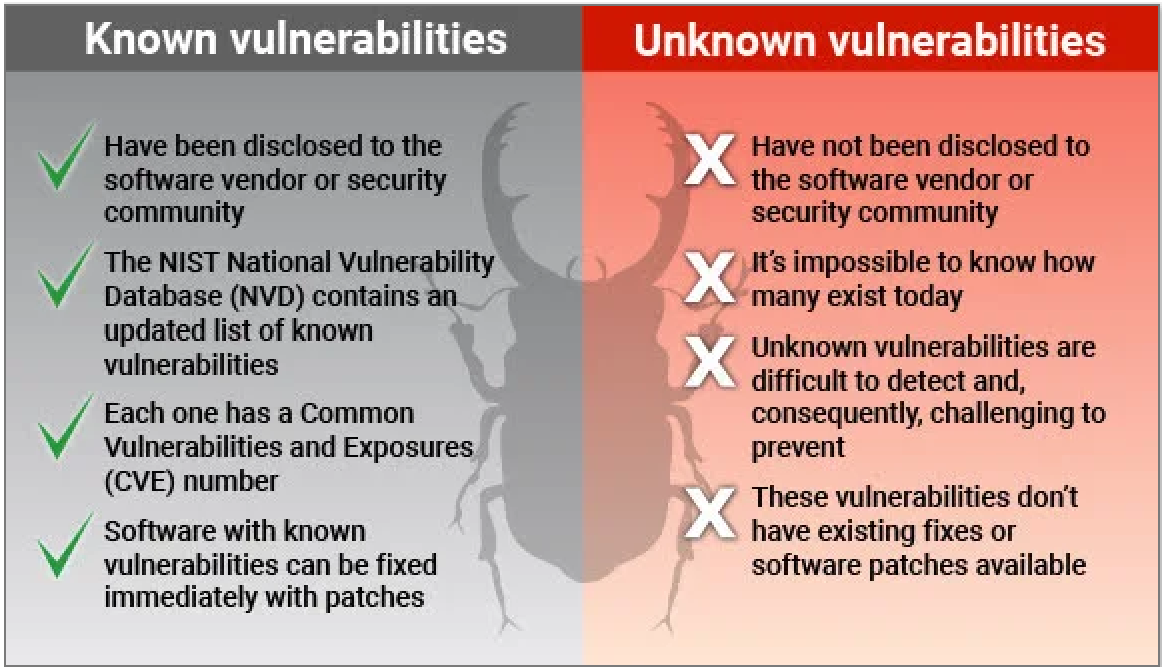
4 Responsible Disclosure

Responsible Disclosure is a model of vulnerability disclosure wherein a vulnerability or an issue is disclosed only after a period of time, allowing concerned parties to produce a patch or mend the vulnerability. Disclosure period is necessary since hardware and software developers often require time and resources to repair their mistakes. But, eventual vulnerability disclosure is necessary to mitigate the feeling of false security and reinforcing responsibility to make the public aware of high impact vulnerabilities. The expected time needed to produce and apply an emergency patch or workaround for the vulnerability may vary between a few days to several months. But generally, software patches are easier to distribute via Internet than hardware patches [19].

Extensive documentation, like NIST National Vulnerability Database (NVD), allows known vulnerabilities to be easily handled or fixed, but unknown vulnerabilities pose a significant security concern. Figure 9 illustrates the difference between Known and unknown vulnerabilities.

There are mixed views on Responsible disclosure. Some researchers might expect financial compensation while reporting vulnerabilities and hence would prefer not disclosing the threat to the public. But, due to social responsibility or financial support of competitive independent firms (in the form of bug bounties) like Facebook, Google, Mozilla, and Barracuda Networks, most players in the commercial vulnerability detection market view Responsible Disclosure as a solution to vulnerability issues.

Following are a few instances where security vulnerabilities were resolved by Responsible Disclosure:



Known vulnerabilities

- ✓ Have been disclosed to the software vendor or security community
- ✓ The NIST National Vulnerability Database (NVD) contains an updated list of known vulnerabilities
- ✓ Each one has a Common Vulnerabilities and Exposures (CVE) number
- ✓ Software with known vulnerabilities can be fixed immediately with patches

Unknown vulnerabilities

- ✗ Have not been disclosed to the software vendor or security community
- ✗ It's impossible to know how many exist today
- ✗ Unknown vulnerabilities are difficult to detect and, consequently, challenging to prevent
- ✗ These vulnerabilities don't have existing fixes or software patches available

Figure 9: Comparison between known and unknown vulnerabilities

- **Starbucks gift card double-spending to create free extra credits** (resolved in 10 days) [20]

  Egor Homakov's blog written on May 21, 2015 details an account of how *race condition* between two simultaneous money transfer between gift cards where he created two $5 transfers from 1st wallet to 2nd wallet containing only $5 and succeded in obtaining 1st wallet with $15 and 2nd wallet with $5 (total $20). He subsequently made a purchase of $16.70 (Figure 10) and deposited $10 extra so that he will not be charged for theft. He managed to get the bug fixed in 10 days after lot of effort.

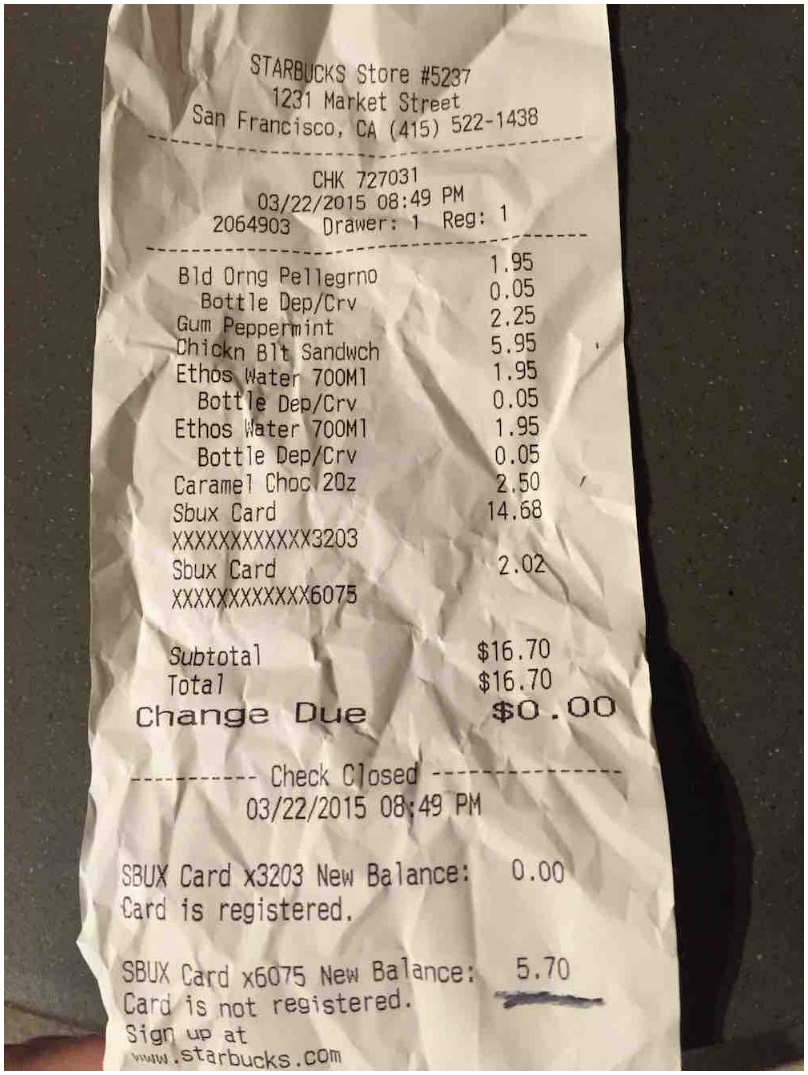
- **Discovery of DNS cache poisoning by Dan Kaminsky** (resolved in 5 months) [23]

  In 2008, Security Researcher Dan Kaminsky presented on a critical Domain Name System (DNS) vulnerability where he described a method to flood the server with large number of DNS queries and the server looks up its cache to respond to these queries instead of verifying the legitimacy of the queries. This allowed attackers to impersonate any legitimate website and steal data.

- **MD5 collision attack for creating false CA certificates** (resolved in 1 week) [18]

  To prevent usage by malicious attackers, the team discovering the MD5 collision attack did not release the software necessary to do the attack and delayed the publication of the improved collision finding techniques. They did not notify the affected Certificate Authorities (CAs) directly before their presentation at Chaos Computer Congress so as to protect internet users. But with Microsoft's offer to act as an intermediary, they were able to contact Verisign and other affected CA. Only 5 hours after their presentation, Verisign stopped using MD5 for all new RapidSSL certificates, successfully eliminating this vulnerability.

- **MBTA vs. Anderson lawsuit** (resolved in 5 months) [21]



STARBUCKS Store #5237
1231 Market Street
San Francisco, CA (415) 522-1438

CHK 727031
03/22/2015 08:49 PM
2064903 Drawer: 1 Reg: 1

Bld Orng Pellegrno	1.95
Bottle Dep/Crv	0.05
Gum Peppermint	2.25
Chickn Blt Sandwch	5.95
Ethos Water 700Ml	1.95
Bottle Dep/Crv	0.05
Ethos Water 700Ml	1.95
Bottle Dep/Crv	0.05
Caramel Choc 2Oz	2.50
Sbux Card XXXXXXXXXXXX3203	14.68
Sbux Card XXXXXXXXXXXX6075	2.02
Subtotal	$16.70
Total	$16.70
Change Due	$0.00

Check Closed
03/22/2015 08:49 PM

SBUX Card x3203 New Balance: 0.00
Card is registered.

SBUX Card x6075 New Balance: 5.70
Card is not registered.
Sign up at
www.starbucks.com

Figure 10: Egor Homakov's receipt for a Starbucks purchase of $16.70 with two gift cards totalling $20

MIT students found vulnerability in the Massachusetts Bay Transportation Authority's (MBTA) CharlieTicket. The lawsuit filed by MBTA in August 2009 prevented the students from presenting their security research at the annual DEF CON hacker convention even though the students had tried contacting MBTA regarding this issue. It took around 5 months for MBTA to completely resolve the vulnerability.

- **The ROCA vulnerability** (resolved in 8 months) [22]

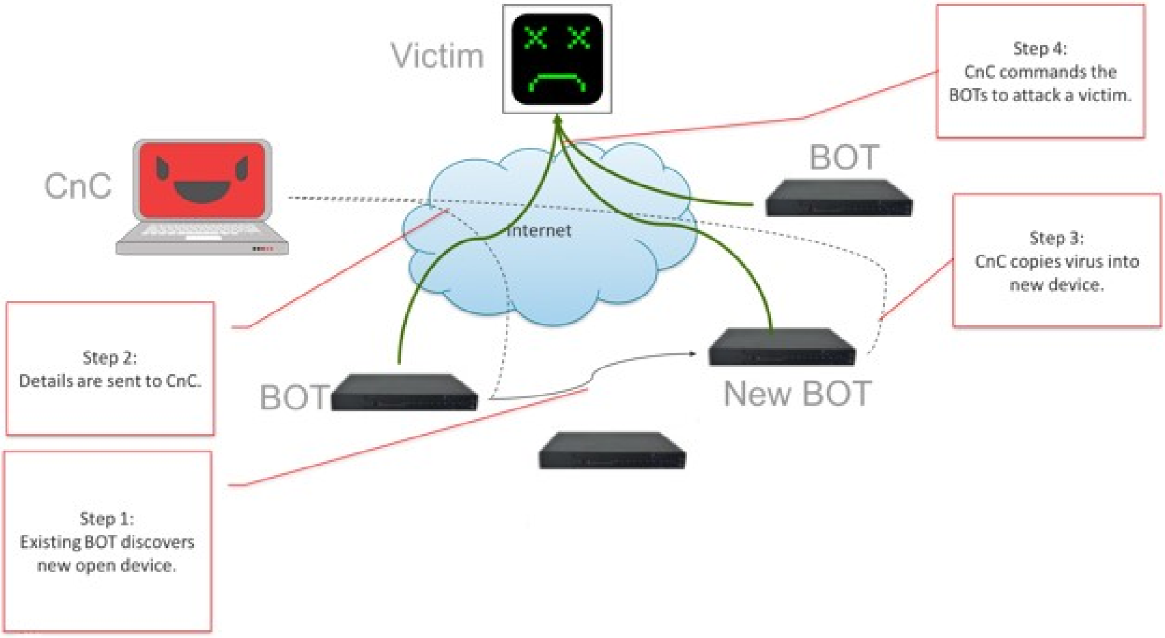
  Discovered in cryptographic firmware used in products made by Infineon Technologies, the ROCA vulnerability was observed in prime-search algorithm used for RSA key generation. This resulted in generation of RSA keys that are relatively cheap and inexpensive to factor. This vulnerability affected the Trusted Platform Modules (TPMs) as well as many smartcards and Hardware Security Modules (HSMs). It took a long time for Infineon Technologies to provide fixes for this bug.

5 Headless IoT Security

Headless system refers to a device or computing system which operates without a monitor, peripheral devices (mouse, keyboard) or graphical user interface (GUI). In most cases they refer to embedded systems that form the backbone for Internet of Things (IoT). Currently there are around 8 billion IoT devices and its estimated that by early 2020s, there will be 25 to 30 billion IoT devices connected worldwide. But there has been lack of concern regarding security challenges of this expansion among many industries involved in this process and it has been predicted that around 25% of the cyber-attacks will target IoT devices by 2020 [27].

Some of the few attacks and incidents targeting headless IoT devices are listed below:

- **DDoS attack by Mirai botnet** (2016) [26, 25]



Victim
Step 4:
CnC commands the BOTs to attack a victim.
CnC
BOT
Internet
Step 3:
CnC copies virus into new device.
Step 2:
Details are sent to CnC.
BOT
New BOT
Step 1:
Existing BOT discovers new open device.

Figure 11: Overview of Mirai System

In August 2016, a self-propagating malware by the name Mirai (Japanese word for *future*), running on the ARC processor and the Linux OS, turned networked devices into remotely controllable *bots* that were used as part of a *botnet* to launch large-scale distributed denial of service (DDoS) attacks on IoT enabled devices like webcams and routers. Its first major attack was on computer security journalist Brian Krebs' web site (20 September 2016) and later it overwhelmed the servers of Dyn, a Domain Name System (DNS) provider causing unavailability of major Internet platforms and services to large sections of users in Europe and North America (21 October 2016). In the next few weeks Mirai disrupted internet service for more than 900,000 Deutsche Telekom customers in Germany, and infected almost 2,400 TalkTalk routers in the UK.

In principle, the Mirai system consists of two main components, the virus and the Command and Control center (CnC). Its working is depicted in Figure 11. The virus actively attacks the current device and the scanner process actively seeks other devices to compromise.

In September 2016, a hacker known as ”Anna-senpai” elected to open-source Mirai's code allowing hackers to develop different strains of Mirai that can take over new vulnerable IoT devices. Finally on December 13, 2017, Paras Jha, Josiah White, and Dalton Norman entered a guilty plea to crimes related to the Mirai botnet.

- **PLC-Blaster** (2016) [24]

  R. Spenneberg, M. Bruggemann, and H. Schwartke designed and demonstrated a method by which Programmable Logic Controllers (PLCs) attacked and compromised each other with an autonomous spread similar to a conventional malware worm at the Black Hat Asia conference in May 2016. As most PLCs are *headless* and thus without a human controllable interface, their work transformed PC or Smart phone controlled attacks to device-to-device managed attacks. This instigated security researchers to look for solutions at both edge level networks and end-to-end networks.

- **TRENDnet Webcam Hack** (2012) [29]

  In January 2012, hackers posted live feeds on the Web from nearly 700 SecurView cameras made by TRENDnet. These were marketed by TRENDnet for various uses ranging from home security to baby monitoring, but they had faulty software which allowed an attacker to obtain the camera's IP address and access live video contents through it. Also, until April 2010, TRENDnet transmitted user login credentials of the cameras in clean, readable text over the Internet via its mobile apps. TRENDnet later made a settlement with the United States Federal Trade Commission and thereafter released a firmware update to rectify the vulnerability, stopped product shipments, and updated all affected models.

- **St. Jude's Hackable Cardiac Devices** (2017) [28]

  United States Food and Drug Administration (FDA) confirmed in January 2017 that St. Jude Medical's implantable cardiac devices (pacemakers, defibrillators, etc) have vulnerabilities in their transmitters which are used for reading the device data and remotely sharing with the physicians. This could allow a hacker to control the device by accessing the transmitter and indulge in malicious activities like depleting the battery or administering incorrect pacing or shocks. St. Jude later developed a software patch to fix the vulnerabilities.

With recent advances and leaps in IoT technologies, security of IoT devices will play an active role in providing both virtual as well as *physical* security where even a simple data breach in device like webcam can cause major information leak. The seriousness of the issue has pushed both companies and legislators to take definitive steps regarding security of headless devices. Notable among these is the effort by Senator Mark Warner to introduce the Internet of Things (IoT) Cybersecurity Improvement Act of 2017 to establish standards for IoT devices purchased by the U.S. government, even though the bill has not yet been passed by the Senate and has only been referred to the Committee on Homeland Security and Governmental Affairs [27]. Many manufacturers have also started taking IoT device security seriously by incorporating it into IoT device design and development, rather than handling it coarsely before shipping.

6 IMSI Catchers

IMSI catcher or **International Mobile Subscriber Identity catcher** is a telephone eavesdropping device that can intercept mobile phone traffic and track the location data of mobile phone users [30]. IMSI catchers demonstrate the principle of Man-in-the-middle (MITM) attack wherein the catcher pretends to be a mobile tower and mediates between the target mobile phones and service provider's real tower. While the GSM (Global System for Mobile Communications) standard (2G) is highly susceptible to MITM attacks through IMSI catchers, current day 3G successor of GSM, the UMTS (Universal Mobile Telecommunications System) standard can be downgraded by sophisticated attacks to non-LTE (Long-Term Evolution) networks rendering the service amenable to interception via IMSI catchers.

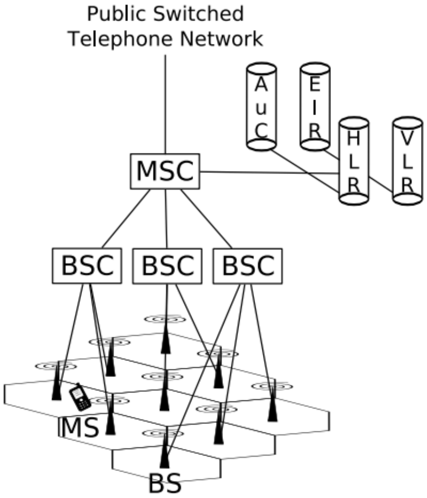


Figure 12: Simplified Architecture of GSM Network

A GSM network architecture (Figure 12) primarily consists of 4 major components - Mobile Stations (MS), Base Stations (BS), Base Station Controllers (BSC) and Mobile Switching Centers (MSC) [31]. A Mobile Station can be identifiable as a mobile phone with a smart card, the Subscriber Identification Module (SIM). An unique 15-digit serial number called the International Mobile Equipment Identity (IMEI) is provided with every mobile phone to prevent stolen phones from accessing the network. Another 15-digit number the International Mobile Subscriber Identity (IMSI) is provided with the SIM to help identify the corresponding subscriber. IMSI consists of the mobile country and network code and subscriber identification number concatenated with each other. The SIM card also stores a 128 bit secret key needed for authentication and key generation. Base Stations enable wireless communication between Mobile Stations and Mobile Switching Centers in a geographical locality and Base Station Controllers are involved in power control and handoff. Finally the Mobile Data Switching Centers consist of many databases like the Home Location Register (HLR), Visitor Location Register (VLR), Authentication Center (AuC), etc which handle routing and authentication procedures [31].

The major weakness of GSM architecture lies in the one-sided authentication of Base Station to a Mobile Station. Figure 13 portrays the Man-in-the-middle attack on GSM network using IMSI catcher. IMSI catcher presents itself as a Base station uses a large signal strength to force nearby Mobile Stations to log in and with special identity request force transmission of IMSI instead of TMSI (Temporary Mobile Subscriber Identity) which was assigned to Mobile Station in initial handshake with the VLR to reduce frequent transmission of IMSI and help in avoiding identification or tracking. IMSI catcher also simultaneously behaves as Mobile Station to the VLR and after authenticating itself, it disables encryption from the Base Station. Thus, it can encrypt the plain text traffic from the Mobile Station and forward to the Base Station [30, 31].

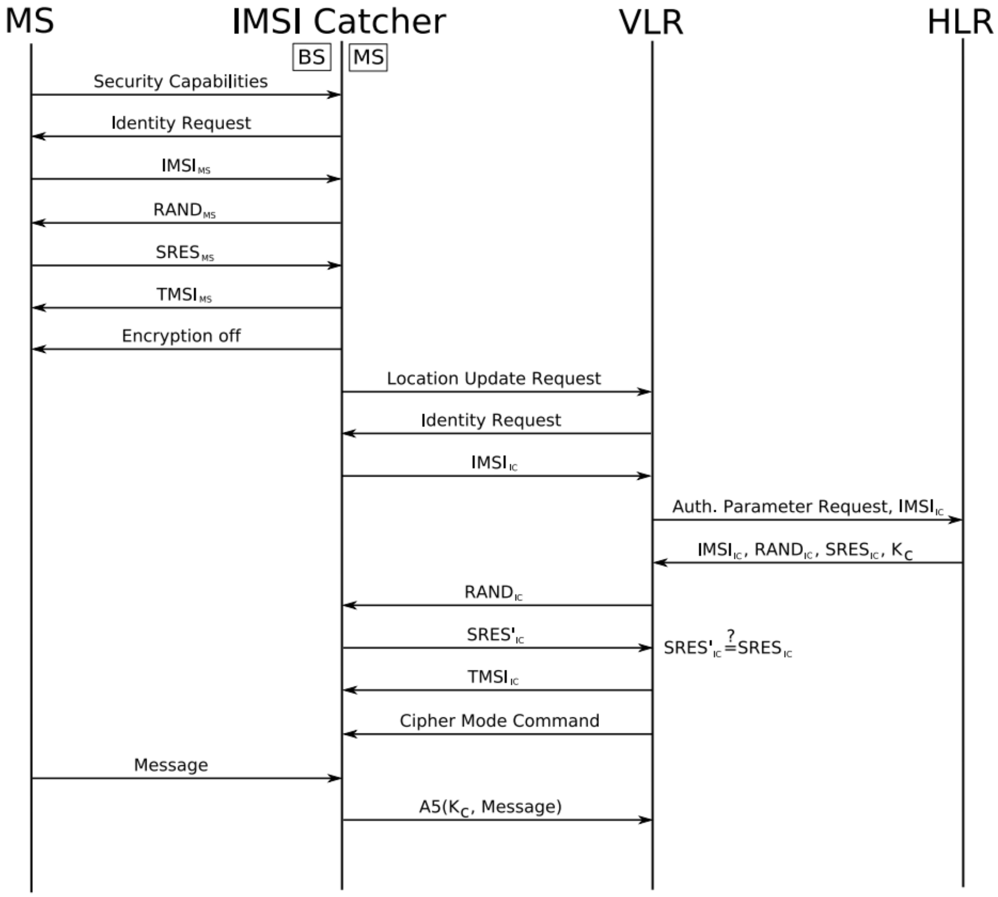


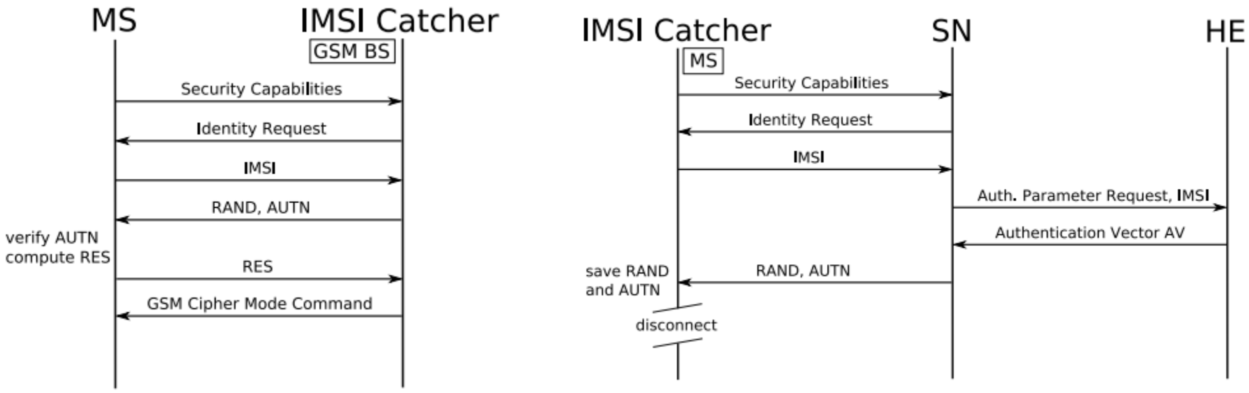
Figure 13: Man-in-the-middle attack using IMSI catcher

IMSI catcher may not be able to accurately localize a Mobile Station, but it can be used the subscriber to verify whether the Mobile Station lies in an area defined by a GSM cell which can range from a radius of a few hundred meters to several kilometers [31].

Recent day UMTS standard improves upon GSM standard by using mutual authentication between Mobile Station and Home Environment (HLR) to prevent MITM attacks and preserves integrity of connection by using a MAC for authentication. But even with its advanced features, Ulrike Meyer and Susanne Wetzel proposed an MITM attack in 2005 by exploiting the interoperation of UMTS with GSM [30]. The attack was executed in three steps:

- Discovering the IMSI or a valid TMSI of the victim which is sent during authentication.
- Impersonation as the victim to the Server Network by sending the IMSI to the Server Network and waiting until a random number RAND and the authentication token AUTN is received before disconnecting. (Figure 14a) which are saved for later use.
- The IMSI Catcher masquerading as a GSM Base Station and sending RAND and AUTN to the victim Mobile Station before the token expires. The attacker then gives the GSM cipher mode command to the Mobile Station (Figure 14b).

With current legal requirements, IMSI catchers allow the police to identify the IMSI and the IMEI of Mobile Stations in case of urgent and definitive suspicion. But safeguarding against IMSI



(a) Attacker obtaining valid authentication token (b) Attacker authenticating as a GSM Base Station

Figure 14: Man-in-the-Middle Attack on UMTS by U. Meyer and S. Wetzel [30]

catchers has been an active research topic with frameworks such as White-Stingray being used to verify popular IMSI Catcher Detector (ISD) apps like SnoopSnitch, Darshak, Cell Spy Catcher, GSM Spy Finder, AIMSICD, etc [32]. With the current state of mobile security, App developers and phone manufacturers should take efficient measures to improve the detection state of the IMSI Catcher Detector apps.

7 SS7 Security

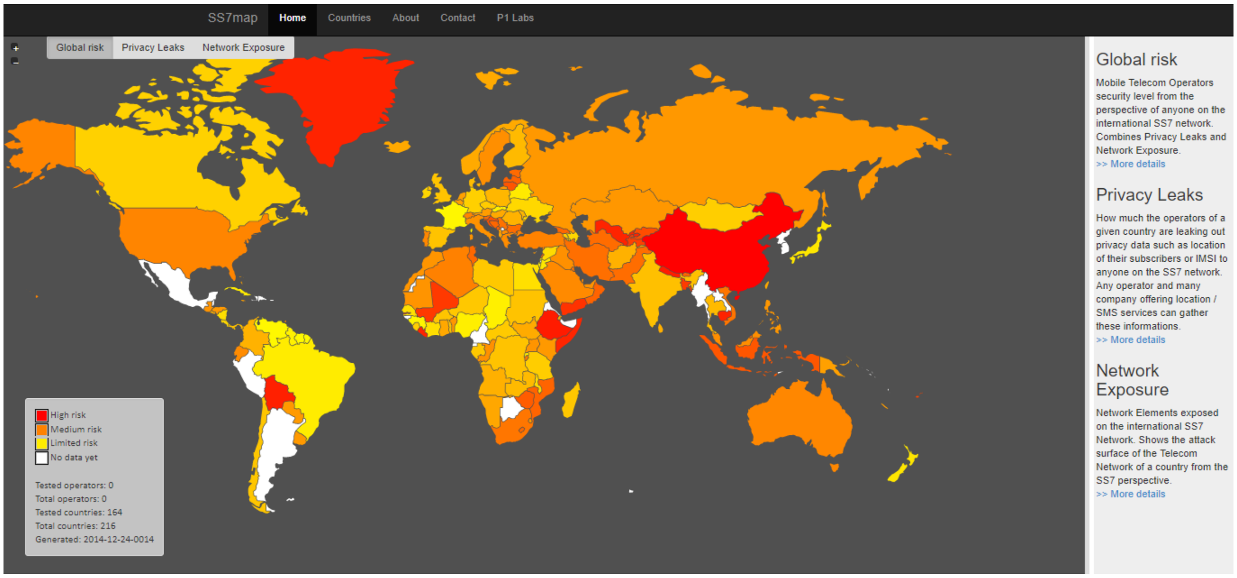
SS7 (Signaling System 7) is a common channel signaling system used in international and local telephone networks [34]. Based on 2G and 3G technologies, it forms the nervous system of Public Switched Telephone Networks (PSTNs) and supports various functionality such as billing, routing, call establishment, and information exchange.

Since decades, SS7 was based upon mutual trust between interconnecting operators and has thus been regarded as a closed trusted network. But today, the signaling network is no longer isolated [35]. Increasing market liberalization and migration to Internet Protocol (IP) has allowed intruders to compromise user's privacy and hamper user experience through call and SMS interception, location tracking, denial of service and fraud (78% of networks were fraud prone [34]). Figure 15 shows the threat levels of SS7 networks across the globe in 2019.

SS7 was standardized by the International Telecommunication Union's Telecommunication Standardization Sector (ITU-T) in 1988 and is used for setting up and ending phone calls, SMS, billing and routing and general information exchange between elements in the GSM and UMTS Core Network. Figure 16 gives an overview of the components of SS7 network where nodes (Signalling Points) are connected by data/signaling links. The three essential nodes used for transferring signals are:

- **Signal Switching Points (SSP)** (nodes A-D) for originating, terminating or switching calls
- **Signal Transfer Point (STP)** (nodes W-Z) for routing signaling messages to destination
- **Signal Control Point (SCP)** (nodes M-P) for performing advanced call processing

SS7 also provides several layers of redundancy in the network to maximize service uptime (for example - deploying STPs and SCPs in pairs).



Tested operators: 0
Total operators: 0
Tested countries: 164
Total countries: 216
Generated: 2014-12-24-0014

Global risk

Mobile Telecom Operators security level from the perspective of anyone on the international SS7 network. Combines Privacy Leaks and Network Exposure.
>> More details

Privacy Leaks

How much the operators of a given country are leaking out privacy data such as location of their subscribers or IMSI to anyone on the SS7 network. Any operator and many company offering location / SMS services can gather these informations.
>> More details

Network Exposure

Network Elements exposed on the international SS7 Network. Shows the attack surface of the Telecom Network of a country from the SS7 perspective.
>> More details

Figure 15: Screen-shot of SS7map, depicting the current state of SS7 security across the globe [37]

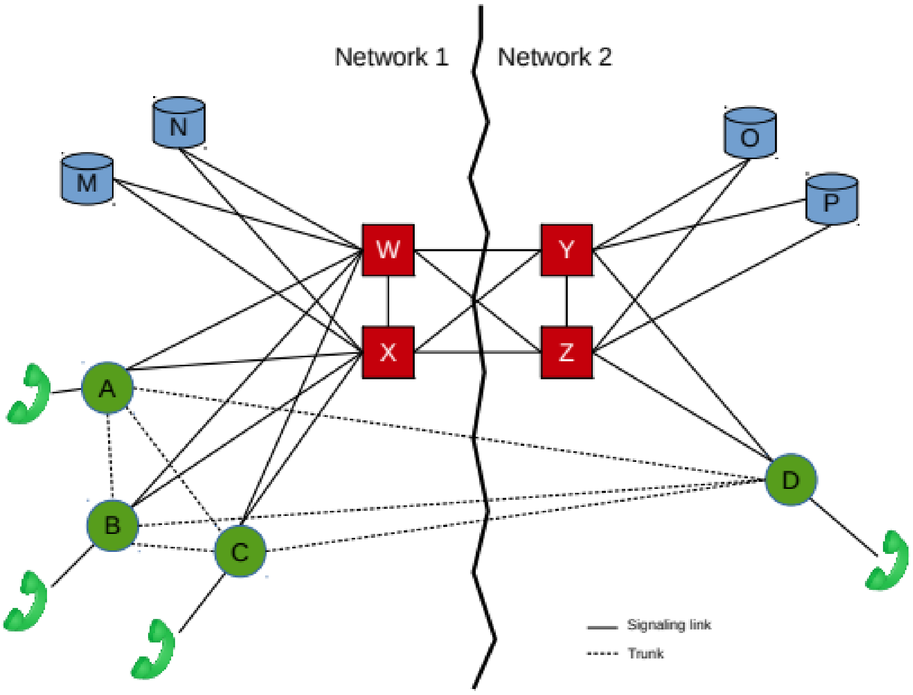
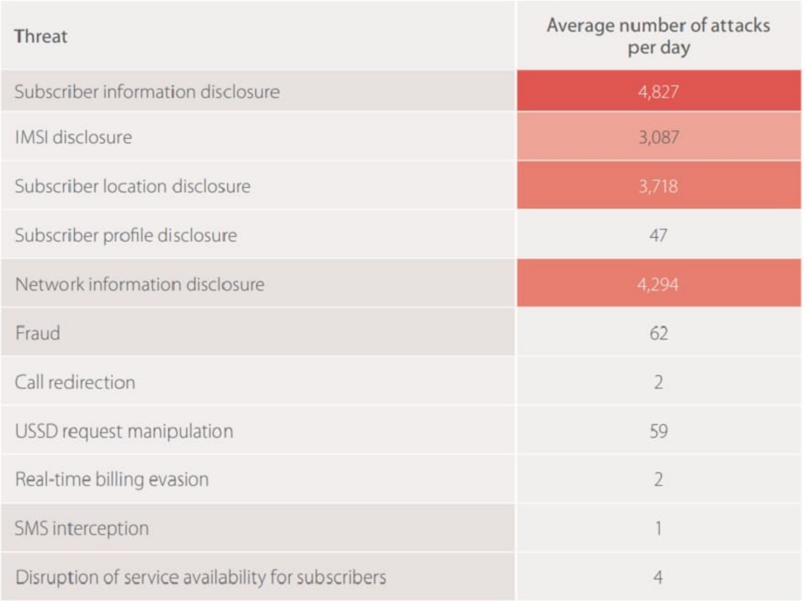


Figure 16: Overview of SS7 network [33]

Attacks on SS7 networks are of primarily four types [35]:

- **Call and SMS Interception** - authentication by *sendIdentification* message, Man-in-the-middle attack by IMSI Catchers, authentication through SMS verification and call forwarding to attackers' device.

- **Location Tracking** - attacker sending *anyTimeInterrogation* (ATI), *provideSubscriberInfo* (PSI), or *provideSubscriberLocation* (PSL) message.
- **Fraud** - abusing Unstructured Supplementary Service Data (USSD) communication.
- **Denial of Service** - removing critical service by using *insertSubscriberData* or *deleteSubscriberData* message or activate call barring for the target.



Threat	Average number of attacks per day
Subscriber information disclosure	4,827
IMSI disclosure	3,087
Subscriber location disclosure	3,718
Subscriber profile disclosure	47
Network information disclosure	4,294
Fraud	62
Call redirection	2
USSD request manipulation	59
Real-time billing evasion	2
SMS interception	1
Disruption of service availability for subscribers	4

Figure 17: Average number of attacks per day by threat types

Recent vulnerabilty exposures in SS7 network (2017) by German newspaper *Süddeutsche Zeitung* reports that hackers obtained a bank customer's username, password, and telephone number and were able to use SS7 vulnerabilities to reroute the two-factor codes and make unauthorized withdrawals from bank accounts. They targeted the German carrier *O2-Telefónica* [36].

Positive Technologies provide many statistics related to attacks on SS7 network [34]. One such statistic is illustrated in Figure 17 where the average number of attacks per day of an operator with a subscriber base of over 40 million people is depicted. From which we can easily infer that attacks with intent of information disclosure are more frequent than other attacks.

Machine learning (ML) and Network Intrusion Detection Systems (NIDS) provide potential solutions to detection of vulnerabilities in SS7 networks. Jensen [33] proposed an **Anomaly-Based Network Abuse Detection System (A-NADS)** to detect anomalies in simulated SS7 networks. These networks were simulated on the SS7 Attack Simulator which can serve as part of a security testbed for further study and research of SS7 vulnerabilities and attacks. Thus considering the current scenario, operators must understand that SS7 is no longer secure and incorporate countermeasures to avoid abuse of users' and subscribers' privacy and maintain integrity of mobile networks.

References

[1] Adam Young and Moti Yung *"Cryptovirology: extortion-based security threats and countermeasures"*, 1996 IEEE conference on Security and Privacy (SP'96), IEEE Computer Society,

Washington, DC, USA, 129-140. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.121.3120&rep=rep1&type=pdf

[2] S. Hou, Y. Ye, Y. Song, and M. Abdulhayoglu, *"Hindroid: An intelligent android malware detection system based on structured heterogeneous information network"*, in 23rd ACM Conference on Knowledge Discovery and Data Mining (SIGKDD), ACM, 2017, pp. 1–9. https://www.cse.ust.hk/~yqsong/papers/2017-KDD-HINDROID.pdf

[3] Robert Lipovský, Lukáš Štefanko, *"Android Ransomware: From Android Defender to DoubleLocker"*, ESET Technical Report, January 2018. https://www.welivesecurity.com/wp-content/uploads/2018/02/Android_Ransomware_From_Android_Defender_to_Doublelocker.pdf

[4] Danny Palmer, *"The nasty future of ransomware: Four ways the nightmare is about to get even worse"*, ZDNet article, October 2017. https://www.zdnet.com/article/the-nasty-future-of-ransomware-four-ways-the-nightmare-is-about-to-get-even-worse/

[5] Arne Maximilian Kaul, *"Ransomware for the Internet of Things"*, Master Thesis, KTH Royal Institute of Technology, 2017. http://www.nada.kth.se/~ann/exjobb/arne_maximilian_kaul.pdf

[6] Ben Dickson, *"The IoT ransomware threat is more serious than you think"*, TechTalks Blog, August 2016. https://bdtechtalks.com/2016/08/22/the-iot-ransomware-threat-is-more-serious-than-you-think/

[7] Ronald A. Kropp, Richard Irving, and Rainer M. Schmitt *"Pulse-rate detection using a fingerprint sensor"*, United States Patent, 2013. https://patents.google.com/patent/US8433110B2/en

[8] Robert Triggs *"How fingerprint scanners work: optical, capacitive, and ultrasonic variants explained"*, Android Authority, 2018. https://www.androidauthority.com/how-fingerprint-scanners-work-670934/amp/

[9] Islam, M. S. *"Heartbeat Biometrics for Remote Authentication Using Sensor Embedded Computing Devices"*, International Journal of Distributed Sensor Networks, 2015. https://doi.org/10.1155/2015/549134

[10] Mahesh Joshi, Bodhisatwa Mazumdar, and Somnath Dey *"Security Vulnerabilities Against Fingerprint Biometric System"* arXiv preprint arXiv:1805.07116, 2018.

[11] Philip Bontrager, Aditi Roy, Julian Togelius, Nasir Memon, and Arun Ross. *"DeepMasterPrints: Generating MasterPrints for Dictionary Attacks via Latent Variable Evolution"*, Proceedings of IEEE International Conference on Biometrics: Theory, Applications and Systems (BTAS), (Los Angeles, USA), October 2018.

[12] Jennifer Schlesinger *"New hacking threats: Fingerprint reader vulnerabilities and sophisticated ransomware"*, CNBC, 2017. https://www.cnbc.com/2017/05/19/new-hacking-threats-fingerprint-vulnerabilities-and-sophisticated-ransomware.html

[13] Brent Sammons *"Breakthrough 3D fingerprint authentication with Snapdragon Sense ID"*, OnQ Blog, 2015. https://www.qualcomm.com/news/onq/2015/03/02/breakthrough-3d-fingerprint-authentication-snapdragon-sense-id

[14] C. Marforio, A. Francillon, and S. Capkun, *"Application collusion attack on the permission-based security model and its implications for modern smartphone systems"*, Technical Report 724, ETH Zurich, April 2011.https://www.research-collection.ethz.ch/handle/20.500.11850/69761

[15] Z. Fang, W. Han, and Y. Li, *"Permission based android security: Issues and countermeasures"*, Computers & Security, 2014. https://www.sciencedirect.com/science/article/pii/S0167404814000261

[16] Chang R, Jiang L, Chen W, et al. *"Towards a multilayered permission-based access control for extending Android security"*, Concurrency Computat Pract Exper, 2018. https://doi.org/10.1002/cpe.4180

[17] Bugiel S, Davi L, Dmitrienko A, Fischer T, Sadeghi A, *"XManDroid: a new Android evolution to mitigate privilege escalation attacks"*, Technische Universität Darmstadt, 2011. https://www.it.iitb.ac.in/frg/wiki/images/e/ec/Xmandroid.pdf

[18] Alexander Sotirov *"Verisign and responsible disclosure"*, Blog, 2009. http://www.phreedom.org/blog/2009/verisign-and-responsible-disclosure/

[19] Stephen A Shepherd *"Vulnerability Disclosure - How do we define Responsible Disclosure?"*, White paper, GIAC SEC Practical, SANS Institute InfoSec Reading Room, 2003. https://www.sans.org/reading-room/whitepapers/threats/paper/932

[20] Egor Homakov *"Hacking Starbucks for unlimited coffee"*, Blog, 2015. https://sakurity.com/blog/2015/05/21/starbucks.html

[21] Michael McGraw-Herdeg and Marissa Vogt *"Students' Subway Security Talk Canceled by Court Order"*, The Tech News, 2009. https://thetech.com/2008/08/08/subway-v128-n30

[22] Nemec M., Sys M., Svenda P., Klinec D., and Matyas V. *"The return of coppersmith's attack: Practical factorization of widely used RSA moduli"*, Proceedings of the 24th ACM SIGSAC Conference on Computer and Communications Security (CCS), 2017. https://crocs.fi.muni.cz/_media/public/papers/nemec_roca_ccs17_preprint.pdf

[23] Thu Pham *"The Great DNS Vulnerability of 2008 by Dan Kaminsky"*, Duo Labs Blog, 2016. https://duo.com/blog/the-great-dns-vulnerability-of-2008-by-dan-kaminsky

[24] R. Spenneberg, M. Bruggemann, and H. Schwartke, *"PLC-blaster: A worm living solely in the PLC"* in Black Hat Asia, Marina Bay Sands, Singapore, 2016. https://regmedia.co.uk/2016/04/29/plc_87458745.pdf

[25] Lily Hay Newman, *"The Botnet That Broke the Internet Isn't Going Away"*, Wired Blog, 2016. https://www.wired.com/2016/12/botnet-broke-internet-isnt-going-away/

[26] Wikipedia contributors, *"Mirai (malware)"*, Wikipedia, The Free Encyclopedia, (accessed January 20, 2019) https://en.wikipedia.org/w/index.php?title=Mirai_(malware)&oldid=877628075

[27] Matt Toomey, *"IoT Device Security is Being Seriously Neglected"*, Aberdeen Blog, 2018. https://www.aberdeen.com/techpro-essentials/iot-device-security-seriously-neglected/

[28] Selena Larson, *"FDA confirms that St. Jude's cardiac devices can be hacked"*, CNNMoney News, 2017. https://money.cnn.com/2017/01/09/technology/fda-st-jude-cardiac-hack/

[29] Richard Adhikari, *"Webcam Maker Takes FTC's Heat for Internet-of-Things Security Failure"*, TechNewsWorld News, 2013. https://www.technewsworld.com/story/78891.html

[30] U. Meyer and S. Wetzel, *"A Man-in-the-Middle Attack on UMTS"*, in ACM Workshop on Wireless Security (WiSe), 2004. https://www.cs.stevens.edu/~swetzel/publications/mim.pdf

[31] Daehyun Strobel *"IMSI catcher"*, Seminar Paper, IT Security Seminar, Ruhr-Universität Bochum, 2007. https://www.emsec.ruhr-uni-bochum.de/media/attachments/files/2011/11/imsi_catcher_update.pdf

[32] S. Park, A. Shaik, R. Borgaonkar, A. Martin, and J.-P. Seifert, *"Whitestingray: Evaluating IMSI catchers detection applications"*, in USENIX Workshop on Offensive Technologies (WOOT), USENIX Association, 2017. https://www.usenix.org/conference/woot17/workshop-program/presentation/park

[33] Kristoffer Jensen *"Improving SS7 Security Using Machine Learning Techniques"*, Master's Thesis, Norwegian University of Science and Technology, 2016. https://brage.bibsys.no/xmlui/bitstream/id/440836/KJensen_2016.pdf

[34] Positive Technologies *"SS7 Vulnerabilities and attack exposure report"*, Annual Report 2018. https://www.ptsecurity.com/ww-en/analytics/ss7-vulnerability-2018/

[35] Hassan Mourad *"The Fall of SS7 – How Can the Critical Security Controls Help?"* White paper, GIAC (GCCC) Gold Certification, SANS Institute InfoSec Reading Room, 2015. https://www.sans.org/reading-room/whitepapers/critical/fall-ss7-critical-security-controls-help-36225

[36] Lily Hay Newman *"Fixing the Cell Network Flaw That Lets Hackers Drain Bank Accounts"*, Wired Blog, 2017. https://www.wired.com/2017/05/fix-ss7-two-factor-authentication-bank-accounts/

[37] P1 Security *"SS7map: SS7 Networks Exposure"*, 2019. http://ss7map.p1sec.com, Accessed on 14/01/2019.