INSE 6120: Cryptographic Protocols and Network Security

Mobile broadband firmware analysis

Tejas Surani(40248859), Nikunj Pathak(40203713), Bhavya Patel(40254222), Abdussamad Syed(40238284), Jenish Patel(40256632), Jaswinder Singh(40258999), Manish Kumar(40230169), Jenil Pansuriya(40256184), Md. Saiduzzaman(40256249), Habib Garba(40189713), Parth Patel(40253832), Hamza Abbas(40256899)

***Abstract*— Mobile baseband firmware plays a critical role in the functioning of cellular devices, governing the communication between mobile devices and cellular networks. An introduction to reverse engineering as it relates to mobile baseband firmware is given in this report, which also explores the goals, strategies, and consequences of such efforts. A detailed analysis of the closed-source code, protocols, and communication methods inherent in mobile baseband firmware is required for reverse engineering. This practice is driven by several goals, such as improving device security, identifying and addressing flaws, and revealing hidden features. The technical difficulties of reverse engineering are examined in the abstract, including the proprietary nature of firmware, encryption techniques, and the ever-changing landscape of cellular technologies. Mobile baseband firmware reverse engineering techniques include hardware debugging tool usage, static and dynamic analysis, and analysis. These methods are employed by researchers to analyze communication protocols, learn more about the firmware's internal workings, and spot possible security holes. The abstract addresses the moral and legal issues that surround reverse engineering operations, emphasizing the fine line that must be drawn between increasing security and possible abuse. Successful reverse engineering initiatives have effects that go beyond the security of a single device and affect more general domains including network security, privacy, and regulatory compliance. Researchers and security professionals can help create stronger security protocols, find and fix vulnerabilities, and ultimately improve the overall resilience of cellular communication systems by learning about the complexities of mobile baseband firmware.**

INTRODUCTION

Welcome to the Mobile Broadband Firmware Analysis Project, a comprehensive exploration into the intricacies of mobile broadband firmware—the pivotal software responsible for processing Layer 2 frames from cellular networks. In this endeavor, participants will navigate the complex landscape of firmware analysis, starting with the challenging task of obtaining firmware images. This involves meticulous research and downloading from manufacturers' websites, typically in the form of firmware updates. The analytical journey unfolds with the imperative to discover the elusive file format and adeptly extract the firmware's core contents.

Once armed with firmware images, participants dive into the heart of the analysis. The process entails identifying and subsequently reverse engineering or decompiling binary code, a task that may necessitate referencing protocol specifications for a more profound understanding. The overarching objective includes ferreting out potential protocol or memory corruption vulnerabilities, a critical aspect of enhancing the overall security and reliability of mobile broadband systems.

Participants are encouraged to undertake a thorough analysis of a minimum of 2 firmware images, with the understanding that the richness of insights derived correlates with the quality of the analysis conducted. Through this project, we aim to foster a deeper understanding of mobile broadband firmware, contributing to advancements in network security and the robustness of communication systems in an ever-evolving technological landscape.

I. WHAT IS GSM?

One of the keystones in the development of mobile communication is the Global System for Mobile Communications, or GSM. GSM has been the industry standard for mobile networks since its introduction in the 1980s, allowing individuals to connect across continents. GSM is still an essential technology today, allowing billions of people worldwide to communicate via voice and data.

A. How GSM Works:

Using a cellular network architecture, GSM divides the world into cells that are serviced by base stations. Both frequency division multiple access (FDMA) and time division multiple access (TDMA) techniques are used by the system. Put more simply, this means that numerous users can utilize the same channel without interfering with one other since each communication is given a distinct frequency and time period.

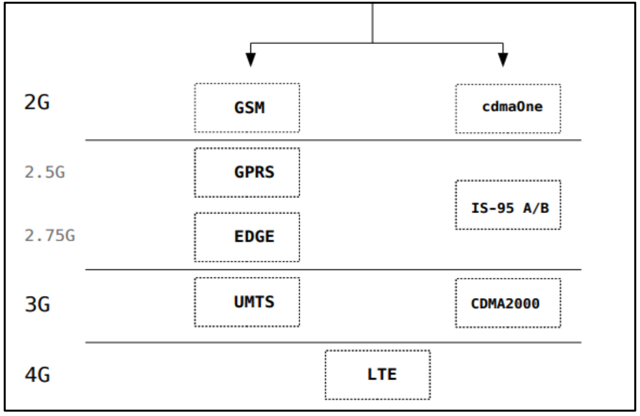


Figure 1. Technology Stack of Cellular Network

Mobile devices communicate with the base station through radio waves, utilizing a specific frequency band. The base station then connects the call or data session through a network of switches and routers, ultimately routing the communication to its destination.

In addition to traditional voice calls, GSM supports a variety of data services, including SMS (Short Message Service) and GPRS (General Packet Radio Service), allowing users to send text messages and access the internet from their mobile devices.

B. Key Features of GSM:

1.Global Reach:

- As the name suggests, one of GSM's defining features is its global reach. GSM networks cover the majority of the world, facilitating international roaming and seamless communication across borders.

2.Interoperability:

- GSM's standardized protocols ensure interoperability between different networks and mobile devices. This interoperability has played a crucial role in creating a cohesive global communication infrastructure.

3.Security:

- GSM incorporates robust security measures to protect user communication. The use of encryption algorithms and authentication mechanisms helps safeguard voice and data transmissions from unauthorized access.

4.Efficiency:

- The TDMA and FDMA techniques employed by GSM enhance spectrum efficiency, allowing multiple users to share the same frequency without compromising the quality of communication.

5.Data Services:

- In addition to voice calls, GSM supports various data services, including SMS, GPRS, and EDGE (Enhanced Data rates for GSM Evolution). This versatility has paved the way for the development of mobile internet and the wide range of applications we use today.

6.SIM Cards:

- GSM introduced the Subscriber Identity Module (SIM) card, a small, portable card that stores user information and allows users to easily switch devices while retaining their identity and services.

GSM has not only revolutionized mobile communication but continues to serve as the foundation for advanced mobile technologies. Its global reach, interoperability, security features, and support for data services have made it an enduring and essential part of the modern communication landscape.

II. MOBILE BASEBAND

Mobile baseband refers to the part of a mobile device responsible for handling communication with cellular networks. It plays a crucial role in enabling voice and data connectivity, managing radio signals, and facilitating the exchange of information between the device and the cellular network infrastructure. The term "baseband" specifically refers to the original frequency range of a transmitted signal before it is modulated to a higher frequency for transmission over the air.

The mobile baseband is an integral component of modern smartphones, ensuring seamless communication between the device and the cellular network. It encompasses various functions, including modulation and demodulation of signals, error correction, channel coding, and protocol management. Essentially, it acts as the bridge between the device's applications and the underlying radio hardware.

One of the key responsibilities of the mobile baseband is to convert digital data from the device into analog signals suitable for transmission over the airwaves. This process involves modulation, where the digital data is combined with a carrier wave to produce a signal that can be transmitted efficiently. On the receiving end, the baseband demodulates incoming signals, extracting the original digital data for the device to process.

Furthermore, the baseband manages the interaction with different cellular network technologies such as 2G, 3G, 4G, and now, 5G. Each generation of mobile networks introduces new challenges and complexities, and the baseband must be adaptable to support these advancements. For example, the transition from 4G to 5G involves not only faster data rates but also changes in network architecture and the use of new frequency bands.

An additional crucial component of mobile baseband functioning is security. To maintain the integrity and confidentiality of network communication, it is in charge of encrypting and decrypting data. Mobile devices contain and transmit sensitive data, including financial transactions and personal information, therefore baseband security measures are critical to preserving user privacy.

In recent years, there has been a growing emphasis on software-defined radio (SDR) in mobile baseband design. SDR allows for greater flexibility and adaptability by enabling updates and improvements through software, reducing the need for hardware modifications. This approach enhances the longevity and upgradability of mobile devices, keeping them compatible with evolving network standards.

In conclusion, the mobile baseband is a critical component of modern mobile devices, enabling communication with cellular networks by managing the transmission and reception of signals. Its multifaceted functions, including modulation, error correction, protocol management, and security measures, ensure a seamless and secure mobile communication experience for users. As technology continues to advance, the adaptability and flexibility of mobile baseband designs become increasingly important to support the evolution of cellular networks.

A. MARKET OVERVIEW:

The mobile baseband market is influenced by factors such as the evolution of cellular network technologies, increasing demand for faster data speeds, and the ongoing development of 5G networks. As consumers expect more advanced features and seamless connectivity, mobile baseband solutions continue to evolve to meet these demands.

B. KEY PLAYERS:

1.Qualcomm:
- One of the leading companies in the mobile baseband market is Qualcomm. Their Snapdragon processors handle multiple cellular technologies, including from 2G to 5G, and frequently contain extensive baseband features. Qualcomm is a leader in the development of solutions for smartphones and other mobile devices thanks to its knowledge of mobile connection.

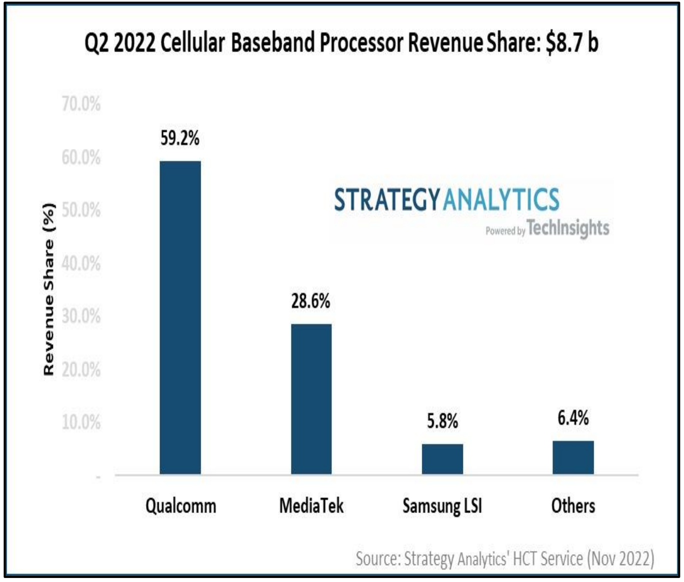
2.Intel (now part of MediaTek):
- Intel was a significant player in the mobile baseband market before selling its smartphone modem business to Apple in 2019. Subsequently, Apple later sold this business to MediaTek. MediaTek is known for providing a range of mobile chipset solutions, including integrated baseband technologies for different network generations.

3.Samsung:
- Samsung, a major player in the smartphone market, also manufactures its Exynos processors, which often include integrated baseband capabilities. Samsung's presence in both mobile devices and semiconductor manufacturing gives it a unique position in the market.

4.Huawei:
- Huawei, a global telecommunications equipment provider, develops its HiSilicon Kirin processors, which include integrated baseband functionalities. While Huawei has faced challenges in certain markets due to geopolitical reasons, it has been a notable player in the mobile baseband sector.

5.MediaTek:
- MediaTek is a Taiwanese semiconductor company that offers a range of mobile baseband solutions. They provide chipsets for a variety of devices, including smartphones, tablets, and other IoT (Internet of Things) devices.

6.Apple:
- While Apple is primarily known for its devices, it has entered the mobile baseband arena through its acquisition of Intel's smartphone modem business. This move allows Apple to develop its in-house baseband solutions for integration into future iPhones and other Apple devices.


Q2 2022 Cellular Baseband Processor Revenue Share: $8.7 b

Source: Strategy Analytics' HCT Service (Nov 2022)

Figure 2. Cellular baseband revenue share

C. Emerging Trends:

1.5G Integration:
- With the rollout of 5G networks worldwide, there is a significant focus on developing baseband solutions that support the higher data rates and low latency offered by 5G technology.

2.Software-Defined Radio (SDR):
- The industry is moving towards more software-centric solutions, allowing for greater flexibility and adaptability through software updates rather than hardware changes.

3.Security Features:
- Given the increasing concerns about cybersecurity, mobile baseband solutions are incorporating enhanced security features to protect user data and privacy.

The mobile baseband market is expected to continue evolving as new generations of cellular technologies emerge, and companies strive to meet the demands of an ever-connected world.

III. HOW BASEBAND IS RELATED TO CELLULAR NETWORKS?

The term "baseband" in the context of cellular networks refers to the original frequency range of a signal before it is modulated to be transmitted wirelessly. The mobile baseband, or cellular baseband processor, is a crucial component in mobile devices that facilitates communication with cellular networks. It plays a fundamental role in the conversion, transmission, and reception of signals in mobile communication. Here's how baseband is related to cellular networks:

1. Signal Modulation and Demodulation:
 - The baseband processor is responsible for converting digital data generated by the mobile device's applications into analog signals suitable for transmission over the airwaves. This process involves modulation, where the digital information is combined with a carrier wave to create a signal that can efficiently traverse the wireless medium.

2. Data Transmission:
 - The modulated signal, now in the radio frequency (RF) domain, is transmitted to the cellular network. The baseband processor manages the transmission of data, ensuring that it conforms to the specific standards and protocols of the cellular network technology in use (2G, 3G, 4G, or 5G).

3. Connection to Cellular Networks:
 - The baseband processor establishes and manages the connection between the mobile device and the cellular network. It handles tasks such as network registration, authentication, and handovers between cell towers as the user moves within the network coverage area.

4. Error Correction and Channel Coding:
 - Cellular networks use various error correction and channel coding techniques to ensure reliable data transmission. The baseband processor is responsible for implementing these techniques, which help in mitigating the effects of signal degradation and interference during wireless communication.

5. Security Protocols:
 - Security is a critical aspect of cellular communication. The baseband processor is involved in encrypting and decrypting data to secure the transmission of sensitive information over the cellular network. This is essential for protecting user privacy and preventing unauthorized access to communication channels.

6.Adaptability to Network Technologies:
 - As cellular network technologies evolve, from 2G to 3G, 4G, and now 5G, the baseband processor must be adaptable to support these changes. It needs to be capable of handling different frequency bands, modulation schemes, and protocol variations to ensure compatibility with the latest network standards.

7. Software-Defined Radio (SDR):
 - The concept of Software-Defined Radio is increasingly being integrated into baseband processors. SDR allows for greater flexibility by enabling updates and improvements through software, reducing the need for hardware modifications. This adaptability is crucial as cellular networks continue to advance.

The mobile baseband is the interface between the digital applications on a mobile device and the cellular network. It manages the conversion, transmission, and reception of signals, ensuring seamless communication while adhering to the specific standards and protocols of the cellular network technology in use. As cellular networks advance, the baseband processor becomes increasingly sophisticated to support higher data rates, lower latency, and the security demands of modern mobile communication.

IV. COMMUNICATION ARCHITECTURE OF MOBILE PHONES

In the early days of mobile phones, the architectural design featured a separation between the Communication Processor (CP) and the Application Processor (AP). This segregation was driven by the limited technology available at the time and the simpler functionality of mobile devices. The CP was dedicated to managing communication functions like cellular data, voice calls, and messaging, while the AP, often considered the device's main "brain," handled application execution and general-purpose processing tasks.

As technology advanced and mobile devices evolved into smartphones, there was a significant shift in the integration trend. Modern smartphones now commonly employ integrated System on Chips (SoCs), consolidating the CP and AP into a single chip. This integration enhances efficiency, reduces power consumption, and contributes to cost-effectiveness in manufacturing.

The CP remains dedicated to managing communication functions, ensuring seamless connectivity to cellular networks and handling the transmission and reception of signals. Meanwhile, the AP continues to serve as the device's main processor, executing applications, running the operating system, and managing user interfaces.

In this integrated architecture, both the CP and AP share access to the device's Random Access Memory (RAM). RAM is a crucial component that provides temporary storage for data while the device is in operation. Both processors utilize RAM for tasks such as buffering data during communication processes, storing application data for quick access, and facilitating smooth multitasking.

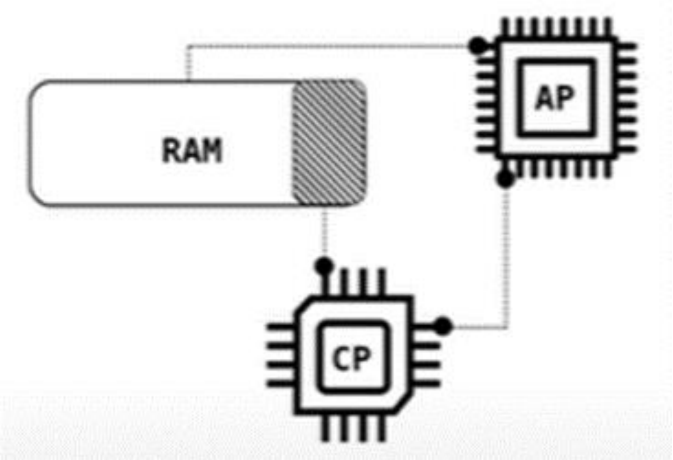


Figure 3. Communication Architecture of Mobile Phones

The consolidation of CP and AP into SoCs represents a technological leap, enabling smartphones to offer a wide range of features, improved performance, and efficient power management. This integration has become a standard in the design of modern mobile devices, reflecting the continuous pursuit of advancements that enhance user experience and device functionality in the ever-evolving landscape of mobile technology.

V. POSSIBLE ATTACKS

The exploitation of the Communication Processor (CP) for unauthorized access represents a significant security concern in the realm of mobile devices. The CP, dedicated to managing telecommunication functions, becomes a potential target for malicious actors seeking unauthorized control over critical aspects of a device's operation. Exploiting the CP can lead to a cascade of security breaches, enabling attackers to intercept communication, manipulate data, and, in a worst-case scenario, gain full control of the device.

A. Exploiting CP for Unauthorized Access:

Code execution on the CP provides attackers with a foothold to compromise telecommunication functions. Once compromised, potential CP-based attacks include:

1. Intercept Communication:
   - Malicious actors can eavesdrop on voice calls and messages, potentially exposing sensitive information.

2. Intercept and Manipulate Data:
   - Attackers may alter SMS contents or reroute messages, leading to misinformation or unauthorized access to sensitive data.

3. Spoofing:
   - Faking the identity of the device or the user in calls or messages can enable attackers to engage in impersonation or fraudulent activities.

4. Denial of Service (DoS):
   - Disrupting communication services can render the device unable to connect, causing inconvenience or financial losses.

B. Escalation to System-Wide Control:

Taking the exploitation a step further involves leveraging Inter-Process Communication (IPC) mechanisms to attack the Application Processor (AP) and gain full control of the device. This escalation to system-wide control includes:

1. Leverage IPC for AP Exploit:
   - Exploiting CP compromise to execute code on the AP, potentially compromising the entire system.
2. Bypass Security Measures:
   - Disabling or circumventing security software via CP to AP pathways, undermining the device's built-in defenses.

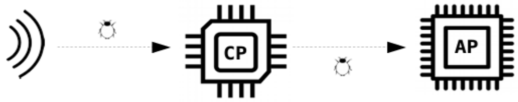


Figure 4. Attack the AP through the IPC mechanisms

C. Full Device Compromise via Advanced Exploits:

In a more advanced scenario, attackers can utilize memory corruption vulnerabilities to exploit the CP and then leverage control over the CP for a complete compromise of the device. This involves:

1. Data Exfiltration:
   - Gaining access to personal data, login credentials, and sensitive files, posing severe privacy and security risks.

2. Persistent Access:
   - Installing rootkits or backdoors for continued control, allowing attackers to maintain unauthorized access over an extended period.

3. Resource Hijacking:
   - Utilizing the compromised device for botnet attacks or cryptocurrency mining, turning it into a resource for further malicious activities.

The defense against such exploits involves robust security measures, regular software updates, and the prompt patching of vulnerabilities. Mobile device manufacturers, software developers, and users alike must remain vigilant to mitigate the evolving threats posed by unauthorized access and exploitation of communication processors.

VI. MEMORY CORRUPTION VULNERABILITY

A Memory Corruption Vulnerability in baseband firmware represents a critical flaw that allows unintended modification of a device's memory, leading to potential system instability or unauthorized access. Such vulnerabilities often arise from coding errors within the complex baseband firmware, including issues like buffer overflows or improper memory handling. The consequences of exploiting these vulnerabilities can be severe, particularly given the central role of the baseband in managing network communication in mobile devices.

The cause of Memory Corruption Vulnerabilities is rooted in coding mistakes that compromise the integrity of the baseband firmware. Buffer overflows, for example, occur when data exceeds the allocated memory space, leading to unintended consequences such as overwriting adjacent memory regions. These errors can result from the intricate nature of the code, which must handle various communication protocols and network interactions.

Exploiting these vulnerabilities grants attackers the ability to intercept or redirect communication data, including phone calls and text messages. This unauthorized access to sensitive information poses significant security risks, as attackers can compromise user privacy, engage in eavesdropping, or manipulate communication content for malicious purposes.

The criticality of Memory Corruption Vulnerabilities in baseband firmware is amplified due to the baseband's pivotal role in managing network communication. Any breach in this area can potentially give attackers control over important features and have a domino effect on the device's overall security. Mitigating such vulnerabilities requires diligent efforts in secure coding practices, regular security audits, and prompt patching of identified flaws to safeguard mobile devices from potential exploitation and ensure the integrity of communication systems.

VII. ENVIRONMENT SETUP

There are two types of environment setup:
1.Dynamic analysis environment
2.Static analysis environment

A. Dynamic Setup

Dynamic Analysis Environment Setup for baseband firmware involves creating a controlled environment to send arbitrary data to the baseband, typically for the purpose of testing and analyzing the behavior of mobile devices. This setup is crucial for security researchers, developers, and analysts seeking to understand and improve the security of baseband firmware.

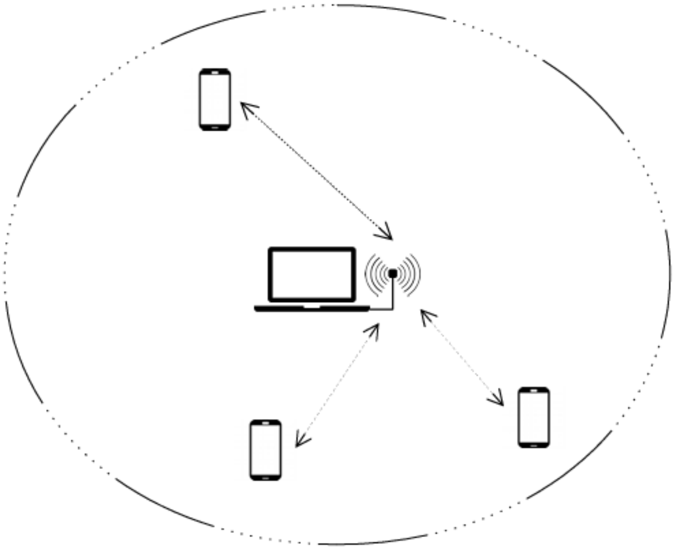


Figure 5. Dynamic Analysis Environment Setup

Software Defined Radio (SDR) and Mobile Network Stack:
To achieve dynamic analysis, a Software Defined Radio (SDR) is employed. SDR is a versatile device capable of transmitting and receiving radio signals. It facilitates the implementation of a mobile network stack or standard in software running on a computer. This enables researchers to emulate a cellular network, allowing them to interact with the baseband firmware in a controlled environment.

Options for SDRs:
Several SDR options are available, catering to various budgets and requirements. Examples include BladeRF x40, BladeRF x115, USRP B200, LimeSDR, and UmTRX. These devices serve as general-purpose transceivers supporting different frequencies and protocols relevant to mobile communications.

1. Software Implementation of Standards:
Complementary to SDRs, software implementations of mobile network standards are necessary. Tools such as YateBTS, OpenBTS, OpenBSC, OpenAirInterface, and OpenLTE provide the infrastructure to emulate different generations of cellular networks. Researchers can choose the software that aligns with their specific testing needs.

2. Provisioned SIM Cards:
Dynamic analysis of baseband firmware often involves the use of provisioned or programmable SIM cards. This is because 3G and 4G networks do not support open authentication, and having control over the SIM cards allows researchers to manipulate authentication processes and simulate various scenarios.

3. Faraday Cage / RF Enclosure:
Operating a cell network without proper authorization is illegal in most countries. To prevent unintentional interference with existing networks and ensure compliance with regulations, researchers use Faraday Cages or RF Enclosures. These structures shield the testing environment from external radio frequency signals, providing a controlled space for dynamic analysis.

The dynamic analysis environment setup for baseband firmware involves combining SDR technology, software implementations of mobile network standards, provisioned SIM cards, and RF enclosures. This enables researchers to study the behavior of baseband firmware under controlled conditions, allowing for the discovery of vulnerabilities, testing of security measures, and development of more robust mobile communication systems. It is essential to emphasize the legal and ethical considerations associated with operating such setups, ensuring that testing activities are conducted responsibly and in compliance with relevant laws and regulations.

B. Static analysis setup

1. Binwalk:
Binwalk, a versatile binary analysis tool, is tailored for firmware analysis, allowing researchers to extract embedded files and code without dynamic execution. It efficiently identifies and extracts various data types within binaries, aiding in detailed firmware breakdowns.

In baseband firmware static analysis, Binwalk extracts file systems, providing critical insights into firmware structure and content. Its extensive detection capabilities cover compressed files, cryptographic signatures, and embedded filesystems, enabling targeted analysis.

For security, Binwalk's static analysis reveals hidden elements, allowing researchers to preemptively identify vulnerabilities in firmware. This proactive approach enhances device security by addressing potential risks before deployment.

Binwalk proves invaluable in static analysis for baseband firmware, offering file extraction, detection capabilities, and security enhancements. This approach ensures robustness and integrity in the evolving cybersecurity landscape.

2. Ghidra:
In the field of firmware static analysis, Ghidra, developed by the National Security Agency (NSA), emerges as a powerful Software Reverse Engineering (SRE) framework. This tool, equipped with advanced software analysis features, is instrumental for security researchers dissecting and comprehending baseband firmware.

Ghidra is a comprehensive framework supporting various processor instruction sets and executable formats, enabling effective disassembly of binary files. Its standout decompilation capabilities convert low-level machine code into higher-level programming constructs, providing analysts with a detailed view of firmware intricacies.

The tool's Graphical User Interface (GUI) is interactive, allowing analysts to manipulate assembly code, graph functions, and visualize code execution flows. This capability is vital for navigating complex codebases, understanding relationships between components, and tracking critical function paths.

Ghidra's open-source nature fosters a thriving community of security professionals and developers. This extensibility enables analysts to tailor their environments, incorporating plugins, scripts, and extensions for a customized and efficient static analysis workflow.

In baseband firmware analysis, Ghidra excels, empowering researchers to navigate codebases, identify vulnerabilities, and understand firmware behavior. Visualization of code execution flow, coupled with decompilation, supports comprehensive documentation for further analysis and security measure development.

Ghidra is a cornerstone in the static analysis setup for baseband firmware, offering decompilation prowess, an interactive GUI, and extensibility through community contributions. As the cybersecurity landscape evolves, Ghidra remains a trusted tool for enhancing the security posture of embedded systems through meticulous static analysis.

VIII. FIRMWARES

In both dynamic and static analysis setups, obtaining the baseband firmware is a crucial first step for researchers and security analysts. It allows for comprehensive examination and understanding of the underlying code to identify vulnerabilities and enhance security measures.

For learning reverse engineering, an initial firmware source is the router firmware of D-Link. The firmware can be accessed at the following link: [D-Link Router Firmware](https://support.dlink.com/resource/PRODUCTS/DAP-1325/REVA/DAP-1325_REVA_FIRMWARE_v1.02B01_BETA.zip). Analyzing router firmware provides a foundational understanding of reverse engineering processes.

Furthermore, for more advanced analysis, the baseband firmware from a Google Pixel 6 device image is obtained. The firmware is extracted from the device image available at: [Google Pixel 6 Device Image](https://developer.android.com/about/versions/14/download). Analyzing mobile device firmware is essential for understanding the intricacies of baseband operations and potential security vulnerabilities.

Lastly, researchers may engage with the Shannon Samsung firmware, available on GitHub at: [Shannon Samsung Firmware](https://github.com/grant-h/ShannonBaseband).
This open-source firmware allows for in-depth analysis and exploration, contributing to the broader knowledge base in the field of baseband security.

Accessing firmware from diverse sources, such as routers, mobile devices, and open-source repositories, provides a rich set of materials for researchers to explore and enhance their skills in both dynamic and static analysis of baseband firmware.

IX. REVERSE ENGINEERING

In the pursuit of in-depth baseband firmware analysis, the utilization of advanced tools such as IDA Pro and Ghidra becomes imperative, surpassing the capabilities of basic tools like Binwalk or Hex Editors. This is especially true for complex architectures like Arm-64, where a higher level of sophistication is required to navigate and comprehend the intricacies of the firmware.

In the initial analysis of the Google Pixel 6 (Arm-64) baseband firmware, the outcomes were limited to a list of disassembled functions. Recognizing the need for more comprehensive tools, the focus shifted to analyzing the Shannon firmware.

Substantial analysis had already been conducted on the Shannon firmware, with additional details available in the provided references. A significant breakthrough occurred when a GitHub repository was discovered, accessible at https://github.com/grant-h/ShannonBaseband. This repository not only contained a list of sample binaries but also provided tools and scripts designed for disassembling these binaries.

It is crucial to note that the tools found in the repository are not applicable to binaries of the Galaxy S6, as they are encrypted. However, the ShannonLoader extension stands out as particularly valuable. This extension automates the recognition of Shannon binaries and dynamically selects the appropriate architecture for disassembly. This level of automation streamlines the analysis process, allowing researchers to focus on understanding the firmware's functionality rather than grappling with manual configuration and adjustments.

The transition from basic tools to advanced platforms like IDA Pro, Ghidra, and specialized extensions like ShannonLoader signifies a commitment to thorough and efficient baseband firmware analysis. The discovery of the ShannonBaseband repository adds a valuable resource to the toolkit, offering not just binaries but also tailored tools and scripts for a more nuanced exploration of Shannon firmware. This approach underscores the importance of leveraging sophisticated tools in the realm of firmware security research.

A. Steps for importing and disassembling binaries using the tools

1. Install the ShannanLoader extension and restart ghidra.
2. After restart, import the binary and make sure to check logs to see if there are any errors.
3. Double click on the binary in the import list to open code browser.
4. Do not auto analyse right away, first run ShannonTraceEntry script – used for identifying debugging infromation and annotation.
5. Now run auto-analyse and wait for it to finish running – takes a long time.
6. Optional but recommended, run the ShannonRename script – helps to name functions based on heuristics.

B. Analysis of firmware

Firmware analysis is a meticulous and time-consuming process that involves dissecting the code embedded in a device's firmware to uncover vulnerabilities, understand functionality, and enhance security. In this analysis, the primary objective was to identify vulnerabilities previously discovered by other researchers, a task that demanded a detailed examination of specific functionalities within the baseband firmware.

The analysis focused on functions implementing critical mobile communication features, including Connection Management (CM), Mobile Management (MM), and Radio Resource Management (RR). Each of these functionalities plays a pivotal role in the seamless operation of mobile devices.

A key challenge in firmware analysis is navigating through the vast codebase to pinpoint relevant functions. The assistance of the ShannonRename script significantly streamlined this process. This script aids in the identification of functions related to Connection Management (CM), Call Control (CC), Short Messaging Service (SMS), Mobile Management (MM), and Radio Resource Management (RR). By leveraging the ShannonRename script, researchers were able to efficiently identify and categorize functions associated with the specified functionalities.

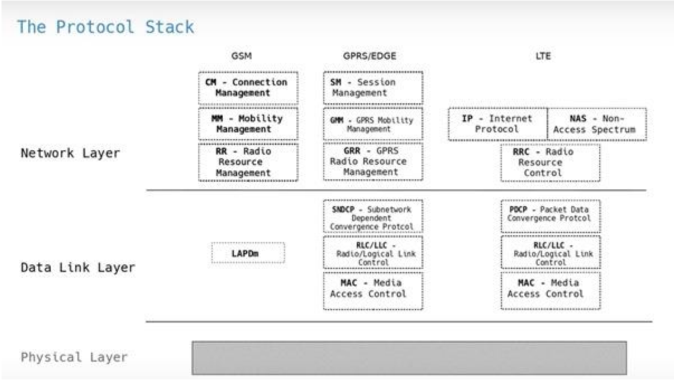

The Protocol Stack
GSM
GPRS/EDGE
LTE
Network Layer
CM - Connection Management
MM - Mobility Management
RR - Radio Resource Management
SM - Session Management
GMM - GPRS Mobility Management
GRR - GPRS Radio Resource Management
IP - Internet Protocol
NAS - Non-Access Spectrum
RRC - Radio Resource Control
Data Link Layer
LAPDm
SNDCP - Subnetwork Dependent Convergence Protcol
RLC/LLC - Radio/Logical Link Control
MAC - Media Access Control
PDCP - Packet Data Convergence Protcol
RLC/LLC - Radio/Logical Link Control
MAC - Media Access Control
Physical Layer

Figure 6. Cellular Protocol Stack

The overarching goal was to ascertain whether the same vulnerabilities identified by previous researchers persisted in the current analysis. This required an exhaustive examination of the functions related to CM, CC, SMS, MM, and RR. The thoroughness of this process is crucial in ensuring the detection of potential security flaws that could pose risks to the device or the network.

By focusing on these specific functionalities, researchers aimed to gain insights into the core operations of the baseband firmware, identifying potential vulnerabilities that could impact connection establishment, call control, messaging services, and resource management. This targeted approach allows for a more efficient and effective analysis, enabling researchers to hone in on critical aspects of the firmware that directly impact the device's communication capabilities.

The analysis of firmware, particularly when focusing on specific functionalities, is a time-intensive yet essential process for uncovering vulnerabilities and ensuring the security robustness of mobile devices. The utilization of tools like the ShannonRename script proves instrumental in accelerating this process, facilitating the identification of functions associated with key functionalities critical to mobile communication. This strategic approach enhances the likelihood of identifying and mitigating potential security risks in the baseband firmware.

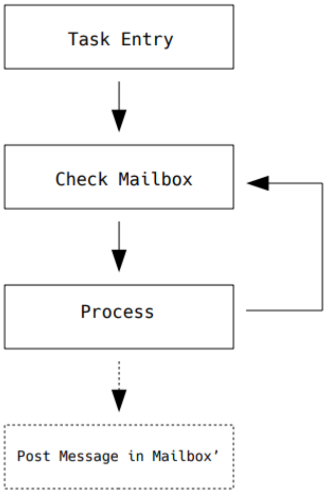
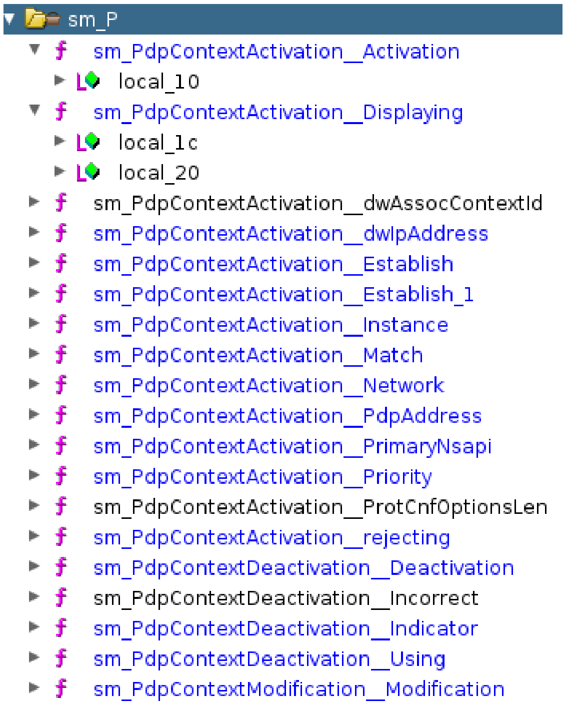


Figure 7. Task Process Loop in Baseband RTOS

In the analysis of baseband firmware functions related to Connection Management (CM), Call Control (CC), Short Messaging Service (SMS), Mobile Management (MM), and Radio Resource Management (RR), researchers are scrutinizing patterns, particularly loops and nested loops. The presence of such patterns may suggest message dequeuing for processing. By examining these coding structures, researchers aim to recognize functions highlighted by previous researchers, potentially uncovering vulnerabilities in the firmware's handling of critical mobile communication functionalities. This meticulous inspection enables the identification of potential weaknesses that may impact the secure operation of the device and its communication processes.

During our analysis we were able to identify functions related to certain cellular processes mentioned by researchers such as the PDP Context Activation (Figure 8) mentioned by Amat Cama, as well as CC Decode (Figure 9) mentioned in the Breaking Band technical presentation.

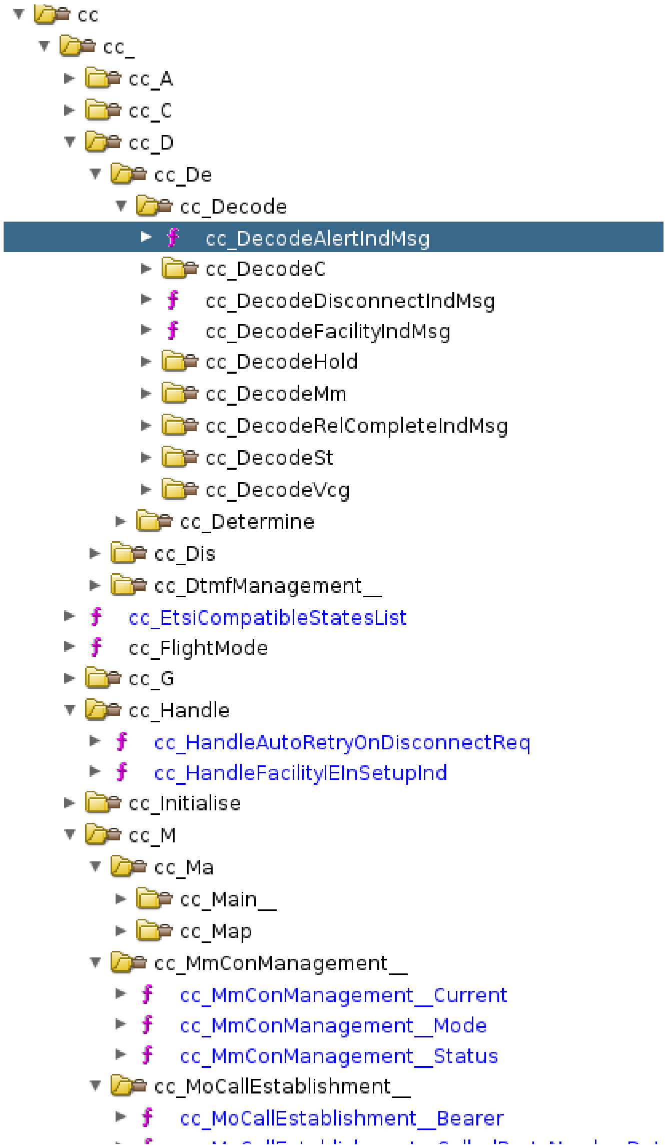


```
▼ sm_P
  ▼ sm_PdpContextActivation__Activation
    ▶ local_10
  ▼ sm_PdpContextActivation__Displaying
    ▶ local_1c
    ▶ local_20
  ▶ sm_PdpContextActivation__dwAssocContextId
  ▶ sm_PdpContextActivation__dwIpAddress
  ▶ sm_PdpContextActivation__Establish
  ▶ sm_PdpContextActivation__Establish_1
  ▶ sm_PdpContextActivation__Instance
  ▶ sm_PdpContextActivation__Match
  ▶ sm_PdpContextActivation__Network
  ▶ sm_PdpContextActivation__PdpAddress
  ▶ sm_PdpContextActivation__PrimaryNsapi
  ▶ sm_PdpContextActivation__Priority
  ▶ sm_PdpContextActivation__ProtCnfOptionsLen
  ▶ sm_PdpContextActivation__rejecting
  ▶ sm_PdpContextDeactivation__Deactivation
  ▶ sm_PdpContextDeactivation__Incorrect
  ▶ sm_PdpContextDeactivation__Indicator
  ▶ sm_PdpContextDeactivation__Using
  ▶ sm_PdpContextModification__Modification
```

Figure 8. PDP Context Activation

In the baseband firmware analysis, functions frequently feature numerous if-else and switch statements, creating a complex code structure. The sheer volume of these statements can make functions indistinguishable from one another. To enhance clarity and facilitate recognition, researchers employ custom labeling of functions and variables. This labeling strategy involves assigning descriptive names to functions and variables, making the code more readable and aiding in the comprehension of intricate relationships within the firmware. Custom labeling is a crucial organizational tool, streamlining the analysis process and contributing to a more efficient understanding of the codebase.

The ShannonRename script operates on heuristics rather than signature-based methods during baseband firmware analysis. Locating the specific function highlighted by researchers becomes challenging due to this heuristic approach. Additionally, the complexity of custom labeling functions and variables adds to the difficulty and time investment required. Successful navigation through this process relies heavily on experience and knowledge, proving essential for effective identification and understanding in the absence of straightforward signatures.



```
▼ cc
  ▼ cc_
    ▶ cc_A
    ▶ cc_C
    ▼ cc_D
      ▼ cc_De
        ▼ cc_Decode
          ▶ cc_DecodeAlertIndMsg
          ▶ cc_DecodeC
          ▶ cc_DecodeDisconnectIndMsg
          ▶ cc_DecodeFacilityIndMsg
          ▶ cc_DecodeHold
          ▶ cc_DecodeMm
          ▶ cc_DecodeRelCompleteIndMsg
          ▶ cc_DecodeSt
          ▶ cc_DecodeVcg
        ▶ cc_Determine
      ▶ cc_Dis
      ▶ cc_DtmfManagement__
    ▶ cc_EtsiCompatibleStatesList
    ▶ cc_FlightMode
    ▶ cc_G
    ▼ cc_Handle
      ▶ cc_HandleAutoRetryOnDisconnectReq
      ▶ cc_HandleFacilityIEInSetupInd
    ▶ cc_Initialise
    ▼ cc_M
      ▼ cc_Ma
        ▶ cc_Main__
        ▶ cc_Map
      ▼ cc_MmConManagement__
        ▶ cc_MmConManagement__Current
        ▶ cc_MmConManagement__Mode
        ▶ cc_MmConManagement__Status
      ▼ cc_MoCallEstablishment__
        ▶ cc_MoCallEstablishment__Bearer
```

Figure 9. CC Decode

X. WORK PERFORMED

The undertaken work involved an exhaustive exploration of diverse resources, including research papers and technical talks, to build a comprehensive understanding of basebands, cellular networks, and related concepts like GSM and GPRS. To gain practical experience, tools such as binwalk and Ghidra were actively utilized. Various firmware binaries were obtained for in-depth analysis.

The analysis process included the application of open-source tools for firmware disassembly, allowing researchers to delve into the code's intricacies. By systematically recognizing functions associated with core GSM functionality, the analysis aimed to uncover potential vulnerabilities. This involved a meticulous examination of the disassembled code, where the utilization of tools like Ghidra played a pivotal role in navigating and comprehending the complexities of the firmware.

Throughout this multifaceted process, the objective was not only to identify functions but also to scrutinize them for possible memory corruption vulnerabilities. The methodology employed a combination of theoretical knowledge, hands-on experience with tools, and practical analysis to unveil potential security risks within the baseband firmware. This comprehensive approach reflects a commitment to a thorough exploration of the firmware's inner workings, aligning with best practices in security research and analysis.

XI. FUTURE WORK

Due to resource and time constraints, dynamic analysis of the baseband firmware was not feasible within the scope of the project. However, for those interested in extending the static analysis, a valuable avenue is exploring FirmWire, a full-system baseband firmware emulation platform designed for fuzzing, debugging, and root-cause analysis. FirmWire provides a dynamic analysis environment for mapping incoming and outgoing packets on the network layer protocol with specific functions, allowing for a more in-depth exploration of potential vulnerabilities. The platform can be accessed through the GitHub repository at https://github.com/FirmWire/FirmWire. Researchers and analysts can leverage FirmWire to conduct dynamic assessments, further enhancing the understanding of the baseband firmware's behavior and potentially uncovering security issues in a controlled environment.

XII. CHALLENGES FACED

The project encountered several challenges, primarily stemming from the starting point of "from scratch," with the team having zero domain knowledge in Baseband and Reverse Engineering. The learning curve was steep, necessitating a comprehensive understanding of baseband concepts, cellular networks, and related technologies.

Locating suitable firmware for analysis posed another hurdle. The scarcity of non-proprietary and unencrypted firmware, essential for meaningful analysis, made the extraction process challenging. The need for high computational power further added to the difficulties, as the extraction of Baseband Firmware demands substantial computing resources.

Additionally, the availability of open-source tools for performing static analysis became a critical concern. The project relied on tools like binwalk and Ghidra, and ensuring their compatibility and effectiveness in the analysis of diverse firmware required extensive exploration.

Despite these challenges, the project's dedication to overcoming these hurdles by learning, researching, and utilizing available resources reflects a resilient approach to baseband firmware analysis. The journey from a knowledge gap to a comprehensive analysis demonstrates the team's commitment to addressing challenges head-on and gaining valuable insights into the intricacies of baseband technology.

XIII. ACKNOWLEDGEMENT

We extend our sincere gratitude to our professor for his invaluable guidance and unwavering support throughout the entirety of our project. His commitment to our success was evident at every stage, as he consistently provided assistance, ensuring clarity and direction in our endeavors. When challenges arose, particularly in the search for appropriate firmware to analyze, our professor's expertise and guidance proved instrumental. He generously dedicated time from his busy schedule each week to engage in meaningful discussions with us, offering valuable insights and advice that greatly enriched our project. His mentorship has been a cornerstone of our academic journey, and we are truly appreciative of the mentorship he provided.

XIV. REFERENCES

- REcon 2016 - Breaking Band (Nico Golde, Daniel Komaromy) - https://www.youtube.com/watch?v=o280NiZjNu8
- A walk with Shannon: A walkthrough of a pwn2own baseband exploit - Amat Cama - https://www.youtube.com/watch?v=6bpxrfB9ioo
- How To Hack Shannon Baseband (from A Phone) by Natalie Silvanovich | hardwear - https://www.youtube.com/watch?v=NnmAikOTHaA
- Reversing & Emulating Samsung’s Shannon Baseband | Grant Hernandez & Marius Muench - https://www.youtube.com/watch?v=ypxgXNtvlgA
- https://www.sstic.org/media/SSTIC2020/SSTIC-actes/how_to_design_a_baseband_debugger/SSTIC2020-Article-how_to_design_a_baseband_debugger-berard_fargues.pdf
- https://github.com/grant-h/ShannonBaseband
- https://github.com/FirmWire/FirmWire

APPENDIX

In the aftermath of our project presentation on November 21st, we endeavored to make significant progress. Regrettably, the challenges posed by upcoming exams and pending coursework in other subjects constrained our ability to allocate the substantial time we had initially envisioned for project advancement. Scheduling collaborative sessions also presented difficulties, as team members grappled with distinct commitments and responsibilities. Despite our best intentions to continue the project's momentum, these external factors hampered our progress during this post-presentation phase. We acknowledge these challenges in the interest of transparency and provide this context in the appendix for a comprehensive understanding of the project's trajectory.