NOTHING SAYS 'SECURITY' LIKE A DOZEN FIREWALLS AND A BIOMETRIC SCANNER

Find more:

Boosty.to

Sponsr.ru

Telegram

Free Issue Section

The perfect starting point for those new to the world of cybersecurity without financial commitment.

Regular Issue Section

Tailored for regular readers who have a keen interest in security and wish to stay abreast of the latest trends and updates.

Pro Issue Section

Designed for IT pro, cybersecurity experts, and enthusiasts who seek deeper insights and more comprehensive resources.

OVERKILL SECURITY

MONTHLY DIGEST. 2024 / 05

Welcome to the next edition of our Monthly Digest, your one-stop resource for staying informed on the most recent developments, insights, and best practices in the ever-evolving field of security. In this issue, we have curated a diverse collection of articles, news, and research findings tailored to both professionals and casual enthusiasts. Our digest aims to make our content both engaging and accessible. Happy reading!

NEWS

Read more: Boosty | Sponsr | TG

DEX & NEXUS

The article details technical aspects of dealing with a specific Android banking trojan, also broader themes in malware analysis, such as the use of obfuscation techniques and the tools available to counteract these methods

📌**String Obfuscation Mechanism:** The Nexus banking trojan uses a string obfuscation mechanism extensively throughout its application code. This complicates the analysis and understanding of the application's functionality.

📌**Analysis Tools:** The analysis mentions the use of both manual decoding and paid tools like the JEB Decompiler for identifying and patching the obfuscated code.

📌**Dalvik Bytecode Inspection:** The case study explores modifying the obfuscated methods by inspecting the Dalvik bytecode, which is part of the DEX files in Android applications.

📌**Tool Release** - dexmod: a tool called dexmod, developed to assist in the patching of Dalvik bytecode that exemplifies how DEX files can be modified to simplify the analysis of Android applications.

📌**Application Permissions:** The analysis of the AndroidManifest.xml file reveals that the trojan requests access to sensitive information such as SMS messages, contacts, and phone calls.

📌**Obfuscated Methods and Patching:** Specific methods like bleakperfect() are highlighted for containing dead code and for their role in decoding strings using XOR operations. The article discusses patching methods to remove redundant code and simplify the analysis.

📌**DEX File Structure:** The case study provides insights into the structure of DEX files, including sections like headers, string tables, class definitions, and method code. It explains how classes and methods are defined and referenced within these files.

📌**Checksum and Signature Updates:** The necessity of updating checksum and SHA-1 signature values in the DEX file's header to ensure content verification is emphasized.

BATBADBUT

📌**Vulnerability:** The critical security vuln is identified as "BatBadBut" and is tracked under CVE-2024–24576

📌**Affected Software:** The vuln exists in the Rust standard library and specifically affects Windows systems

📌**Severity Rating:** It has been given the highest severity rating with CVSS score of 10.0, indicating maximum severity

📌**Vulnerability Details:** The flaw arises from the Rust standard library not properly escaping arguments when invoking batch files on Windows using the Command API. This could allow an attacker to execute arbitrary shell commands by bypassing the escaping

📌**Conditions for Exploitation:** Successful exploitation requires specific conditions: execution of a command on Windows, the command does not specify the file extension or uses .bat or .cmd, the command contains user-controlled input as part of the command arguments, and the runtime fails to escape the command arguments properly for cmd.exe

📌**Affected Versions:** All versions of Rust before 1.77.2 on Windows are impacted by this vulnerability

📌**Broader Impact:** The vulnerability also affects other programming languages, including Erlang, Go, Haskell, Java, Node.js, PHP, Python, and Ruby, though not all have released patches

📌**Mitigation Recommendations:** Users are advised to move batch files to a directory not included in the PATH environment variable to prevent unexpected execution. Developers should upgrade to Rust version 1.77.2 to patch the vulnerability

📌**Discovery & Reporting:** The vulnerability was discovered by a security engineer from Flatt Security known as RyotaK and reported to the CERT/CC

📌**Response from Rust:** The Rust Security Response Working Group acknowledged the issue and has since improved the robustness of the escaping code and modified the Command API to return an InvalidInput error if an argument cannot be safely escaped

📌**Other Languages' Response:** Patches released by maintainers of Haskell, Node.js, PHP, and yt-dlp to address the command injection bug

VULNERABILITIES IN LG'S WEBOS / LG SMARTTV

Security researchers from Bitdefender have identified multiple vulnerabilities in LG's WebOS, affecting various models of the company's smart TVs. These vulnerabilities, if exploited, could allow attackers to gain unauthorized root access to the devices.

Affected Versions and Models:

📌The vulnerabilities impact LG TVs running WebOS versions 4.9.7 to 7.3.1 across models such as LG43UM7000PLA, OLED55CXPUA, OLED48C1PUB, and OLED55A23LA

Specific Vulnerabilities:

📌**CVE-2023-6317**: Allows attackers to bypass PIN verification and add a privileged user profile without user interaction

📌**CVE-2023-6318:** Enables attackers to elevate their privileges and gain root access

📌**CVE-2023-6319:** Permits operating system command injection by manipulating a library for displaying music lyrics

📌**CVE-2023-6320:** Allows for the injection of authenticated commands by exploiting the com.webos.service.connectionmanager/tv/setVlanStaticAddress API endpoint

Discovery and Reporting:

📌These vulnerabilities were discovered by Bitdefender in November 2023 and reported to LG, which subsequently released patches on March 22, 2024

Scope of Impact:

📌Over 91,000 devices have been identified as potentially vulnerable. These devices are primarily located in South Korea, Hong Kong, the US, Sweden, and Finland

Mitigation and User Action:

📌LG has released patches for these vulnerabilities, which are available through the TV's settings menu under Software Update

📌Users are advised to enable automatic software updates to ensure their devices receive the latest security patches

Potential Risks:

📌If exploited, these vulnerabilities could allow attackers to take control of the TV, access sensitive user data, and potentially use the compromised device as part of a botnet or for other malicious activities

Security Recommendations:

📌Besides applying the latest firmware updates, users should use strong, unique passwords for their devices and secure their Wi-Fi networks to further reduce the risk of exploitation

TA547 PHISHING CAMPAIGN

The TA547 phishing campaign using the Rhadamanthys stealer represents a significant evolution in cybercriminal tactics, notably through the integration of AI-generated scripts. This development serves as a critical reminder for organizations to continuously update and adapt their cybersecurity strategies to counter sophisticated and evolving threats.

Key Details of the Attack

📌**Impersonation and Email Content:** The phishing emails were crafted to impersonate the German company Metro AG, presenting themselves as invoice-related communications. These emails contained a password-protected ZIP file, which when opened, triggered a remote PowerShell script

📌**Execution Method:** The PowerShell script executed directly in memory, deploying the Rhadamanthys stealer without writing to the disk. This method helps avoid detection by traditional antivirus software

📌**Use of AI in Malware Creation:** There is a strong indication that the PowerShell script was generated or at least refined using a large language model (LLM). The script featured grammatically correct and highly specific comments, which is atypical for human-generated malware scripts

Evolving Tactics and Techniques

📌**Innovative Lures and Delivery Methods:** The campaign also experimented with new phishing tactics, such as voice message notifications and SVG image embedding, to enhance the effectiveness of credential harvesting attacks

📌**AI and Cybercrime:** The use of AI technologies like ChatGPT or CoPilot in scripting the malware indicates a significant shift in cybercrime tactics, suggesting that cybercriminals are increasingly leveraging AI to refine their attack methods

📌**Broader Implications:** This campaign not only highlights the adaptability and technical sophistication of TA547 but also underscores the broader trend of cybercriminals integrating AI tools into their operations. This integration could potentially lead to more effective and harder-to-detect cyber threats

Recommendations for Defense

📌**Employee Training**: Organizations should enhance their cybersecurity defenses by training employees to recognize phishing attempts and suspicious email content

📌**Technical Safeguards:** Implementing strict group policies to restrict traffic from unknown sources and ad networks can help protect endpoints from such attacks

📌**Behavior-Based Detection:** Despite the use of AI in crafting attacks, behavior-based detection mechanisms remain effective in identifying and mitigating such threats

FBI IC3

Attackers are employing a variety of methods, including phishing emails with malicious attachments, obfuscated script files, and Guloader PowerShell, to infiltrate and compromise victim systems. Invoice fraud, a form of business email compromise (BEC), is one of the popular methods used by attackers to deceive victims. In this type of scam, a third-party requests payment fraudulently, often by impersonating a legitimate vendor

Invoice scams pose a significant threat to businesses, as they can result in substantial financial losses and irreparable damage. According to the FBI IC3 report, in 2022, BEC attacks caused $2.7 billion in losses to US victims, making it the most pervasive form of business email compromise

Some indicators of fraudulent email invoices include requests for personally identifiable information (PII), unusual requests such as changes to banking or payment information, and invoices with unusual dollar amounts. Additionally, attackers often use obfuscation techniques to evade defenses and make their malicious activities more difficult to detect.

TeleTracker

TeleTracker offers a suite of tools for threat intelligence analysis, focusing on Telegram channels used for malicious purposes. Its features facilitate the monitoring and disruption of active malware campaigns, making it a valuable resource for cybersecurity professionals. These scripts are particularly useful for threat intelligence analysts or researchers aiming to monitor, collect, and track adversaries using Telegram for command and control (C2) communications.

Features

📌**View Channel Messages & Download Content:** Allows users to view messages within a channel and download content directly to a newly created 'downloads' folder in the current working directory. It supports the download of various file types including documents, photos, and videos.

📌**Send Documents via Telegram:** Users can optionally send messages and documents through Telegram, supporting all Telegram file types with auto-detection of MIME type.

📌**Message Selection:** Provides the option to select a specified number of messages or a specific message_id for download, with downloads always occurring from the newest to the oldest message.

📌**Log Saving:** Saves logs in a pretty text format with basic information under a file named <bot_name>.txt.

Usage

📌To send a message to TG channel, use the command: python TeleTexter.py -t YOUR_BOT_TOKEN -c YOUR_CHAT_ID -m "Your message here"

📌For continuous message sending (spamming), add the --spam flag to the command.

📌TeleViewer.py is the latest tool allowing users to view and download all messages and media from a threat actor-controlled Telegram channel. This feature can be accessed by selecting the number 6 from the initial menu after running TeleGatherer.py.

Abusing WSUS with MITM to perform ADCS ESC8 attack

This article serves as a technical guide on how a combination of network sniffing, MITM attacks, and exploitation of ADCS can lead to significant security breaches, emphasizing the need for robust security measures in network configurations and certificate handling processes.

📌**WSUS Configuration and Vulnerability:** The article details how a Windows Server Update Services (WSUS) server, configured to work over HTTP, can be exploited. The WSUS server's protocol configuration is accessible by querying a specific registry key. This setup allows for the potential sniffing of traffic using tools like Wireshark, which can capture the communication between clients and the WSUS server.

📌**MITM Attack Execution:** The core of the attack involves a Man-in-the-Middle (MITM) approach where an attacker intercepts and relays requests from a client machine to the WSUS server. During this process, the attacker can manipulate the communication to redirect requests to a rogue server or manipulate the responses.

📌**ADCS ESC8 Exploit:** The intercepted communication is then used to facilitate an Active Directory Certificate Services (ADCS) ESC8 attack. This involves relaying the intercepted requests to a Certificate Authority web enrollment page to request a certificate using a compromised computer's credentials. Successfully executing this attack can allow the attacker to obtain unauthorized certificates that can be used for further attacks within the network.

📌**Toolset:** PKINITtools and scripts for manipulating Kerberos tickets and exporting them for use in the attack help in extracting and utilizing the credentials from the intercepted traffic to authenticate against the ADCS and request certificates.

📌**Security Implications and Recommendations:** The attack demonstrates a significant security risk in using unsecured protocols (HTTP) for critical infrastructure like WSUS and ADCS. The article suggests that securing these communications using HTTPS and implementing strict access controls and monitoring could mitigate such attacks.

Passkeys: A Shattered Dream

The blog post provides a critical perspective on the implementation and user experience of Passkeys, particularly in the context of WebAuthn (Web Authentication). The author shares a personal anecdote to highlight the issues faced by users, leading to a broader critique of Passkeys.

📌**Personal Experience with Passkey Failure:** The author begins with a personal story where their partner was unable to access their home light control system because her Apple Keychain had deleted the Passkey she was using. This incident serves as an example of the practical issues users face with Passkeys.

📌**Critique of WebAuthn's Evolution:** The author reflects on their involvement with WebAuthn, starting from its early days. They recount their optimism and contributions to the WebAuthn workgroup, aiming to improve the standard. However, they express disappointment in how technology has evolved, particularly criticizing the concept and implementation of Passkeys.

📌**Passkeys as a Platform Lock-in Tool:** The article argues that Passkeys, rather than being a solution for secure and user-friendly authentication, have become a means for platforms to lock users into their ecosystems. The inability to extract or export credentials is highlighted as a significant drawback, leading to what the author describes as "long term entrapment of users."

📌**User Experience Concerns:** The author shares their partner's negative experience with Passkeys, noting her preference to return to traditional passwords for their simplicity and reliability. This sentiment is echoed by the author, who reluctantly admits that password managers offer a better user experience than Passkeys.

📌**Conclusion and Reflection:** The author concludes by expressing a sense of disillusionment with Passkeys, suggesting that the initial promise of a secure and user-friendly authentication method has been compromised. They hint at the irony of releasing a new version of their WebAuthn library for Rust amidst these reflections.

Read more: Boosty | Sponsr | TG

LockBit publishes confidential data stolen from Cannes hospital in France

📌LockBit is the most dangerous ransomware in the world and has been responsible for a significant number of attacks in France between April 2022 and March 2023.

📌LockBit accounted for 57% of known attacks in France during this period, which is significantly higher than its nearest competitor, ALPHV.

📌The number of monthly attacks in France has been highly volatile, with LockBit being responsible for the majority of this volatility.

📌The French economy is large enough to provide a fertile hunting ground for cybercriminals, and it is possible that some of LockBit's affiliates have decided to specialize in attacking French targets.

📌In July 2022, La Poste Mobile, a mobile carrier owned by French postal company La Poste, suffered a LockBit ransomware attack, resulting in the publication of private information of more than a million and a half people in France.

📌In August 2022, attackers demanded $10 million after a ransomware attack on the Center Hospitalier Sud Francilien (CHSF), a 1000-bed hospital near Paris, causing disruption to computer systems and resulting in patients having to be sent elsewhere and surgeries being postponed.

📌In mid-November 2022, French defense and technology group Thales confirmed a data breach affecting contracts and partnerships in Malaysia and Italy, with the perpetrators using LockBit ransomware.

📌France was the fifth most attacked country in the world between April 2022 and March 2023, with the government sector being attacked more often than in similar countries.

📌The reasons for LockBit's dominance in France are unclear, but it may be due to the group's ability to exploit opportunities outside of the Anglosphere and the possibility that some of its affiliates have specialized in attacking French targets.

📌LockBit operates as a Ransomware-as-a-Service (RaaS) model, with attacks being carried out by independent criminal gangs, referred to as "affiliates", who pay the LockBit gang 20% of the ransoms they extract.

📌The true number of LockBit attacks is likely far higher than the number of known attacks, as many victims choose to pay the ransom rather than risk having their data published on the dark web.

📌LockBit has been linked to attacks on hospitals, governments, and businesses globally, causing significant harm to thousands of victims.

📌Law enforcement agencies have been working to disrupt LockBit's operations, with several people alleged to be linked to the gang arrested in Ukraine and Poland.

📌Despite these efforts, LockBit continues to operate and launch attacks, with the group's purported leader vowing to continue their activities.

📌The U.S. State Department has announced monetary rewards of up to $15 million for information that could lead to the identification of key leaders within the LockBit ransomware group and the arrest of any individual participating in the operation.

📌Since January 2020, LockBit actors have executed over 2,000 attacks against victims in the United States and around the world, causing costly disruptions to operations and the destruction or exfiltration of sensitive information.

📌More than $144 million in ransom payments have been made to recover from LockBit ransomware events.

📌In response to the ransom demand, CHC-SV stated, "Public health establishments never pay ransom in the face of this type of attack."

📌The hospital also promised to notify patients and stakeholders if the ransom gang decided to publish any stolen data.

📌At the time of this report, there has been no statement from the Hôpital de Cannes regarding the alleged published data

Genzai. The IoT Security Toolkit

The GitHub repository for Genzai, developed by umair9747, is focused on enhancing IoT security by identifying IoT-related dashboards and scanning them for default passwords and vulnerabilities.

📌**Purpose and Functionality:** Genzai is designed to improve the security of IoT devices by identifying IoT dashboards accessible over the internet and scanning them for common vulnerabilities and default passwords (e.g., admin:admin). This is particularly useful for securing admin panels of home automation devices and other IoT products.

📌**Fingerprinting and Scanning Process:** The toolkit fingerprints IoT products using a set of signatures from signatures.json. After identifying the product, it utilizes templates stored in its databases (vendor-logins.json and vendor-vulns.json) to scan for vendor-specific default passwords and potential vulnerabilities.

📌**Supported Devices and Features:** As of the last update, Genzai supports fingerprinting over 20 different IoT-based dashboards. It also includes templates to check for default password issues across these dashboards. Additionally, there are 10 vulnerability templates available, with plans to expand this number in future updates. Some of the IoT devices that can be scanned include wireless routers, surveillance cameras, human-machine interfaces (HMI), smart power controls, building access control systems, climate controls, industrial automation systems, home automation systems, and water treatment systems.

📌**Updates and Contact Information:** The repository indicates that Genzai is an actively maintained project, with plans for adding more vulnerability templates in the coming updates.

UserManagerEoP / CVE-2024-21447

The UserManager EoP exploit by Wh04m1001 targets a vulnerability identified as CVE-2023-36047, which was later tracked as CVE-2024-21447 after additional fixes by Microsoft.

UserManager EoP Exploit

📌**Vulnerability Discovery:** The exploit was discovered by the repository owner last year and affects the UserManager service in Windows.

📌**Nature of Vulnerability:** The flaw involves the UserManager service improperly copying files from a directory that can be controlled by a user, leading to an elevation of privilege (EoP).

📌**Partial Fix and Re-exploitation:** Initially, Microsoft addressed only the write aspect of the file copy operation. However, the read operation continued to be executed with NT AUTHORITY\SYSTEM privileges, which was not secured in the first patch.

📌**Exploit Mechanism:** The exploit takes advantage of the unsecured read operation to access critical system files like SAM, SYSTEM, and SECURITY hives from a shadow copy.

📌**Final Resolution:** The vulnerability was fully addressed by Microsoft recently and is now cataloged under a new identifier, CVE-2024-21447.

Code Analysis

The GitHub repository contains exploit code that demonstrates how to manipulate the UserManager service's file handling to escalate privileges.

📌**Identifying Vulnerable Operations:** Code to identify and target the specific vulnerable read operation performed by the UserManager.

📌**Exploiting the Flaw:** Scripts or commands that manipulate the file operations to redirect or access unauthorized data.

📌**Leveraging System Privileges:** Utilizing the elevated privileges gained from the exploit to perform unauthorized actions, such as accessing or modifying system files and settings.

Architecture of NES Consoles

It seems you've traded the thrilling world of social interactions for the captivating realm of game console research. Let's dive into the depths of your newfound obsession called the Super Nintendo Entertainment System (SNES)? Fabien Sanglard, our hero, has meticulously dissected the SNES, offering us a trilogy of articles that could very well replace any human interaction.

First off, we have the exposé on SNES cartridges, those magical plastic blocks that, surprise, held more than just the dreams of 90s kids. They were technological marvels with their own hardware, including the oh-so-essential CIC copy protection chip.

Then, the author takes us on a historical journey through the evolution of the SNES motherboard. Twelve versions over twelve years, each one reducing the number of chips and components.

And let's not forget the heartwarming tale of the SNES's clock generators. These little timekeepers made sure everything ran like clockwork (pun absolutely intended). Because what's a gaming console without its precise timing to keep those tool-assisted speedruns accurate? It's not like gamers have anything better to do, like, say, going outside.

So, there you have it, a trilogy of articles that could very well serve as a substitute for human interaction. Who needs friends when you have the intricate details of the SNES to keep you warm at night? Thank you, Fabien Sanglard, for giving us the perfect excuse to avoid social obligations in favor of gaming console research.

SNES Cartridges:

The SNES cartridges were unique in that they could include additional hardware such as the CIC copy protection chip, SRAM, and enhancement processors like the "Super Accelerator 1" (SA-1). These processors significantly boosted the console's capabilities, allowing for advanced graphics and gameplay features. It highlights the evolutionary steps Nintendo took with the SNES motherboard to enhance the system's efficiency and cost-effectiveness over time.

Key Features

📌The SNES motherboard underwent significant changes throughout its production, primarily aimed at reducing the complexity and cost of the system.

📌The motherboard started with a high number of chips and components which were gradually reduced in later versions.

Chip Reduction

📌One of the major advancements in the SNES motherboard design was the introduction of the 1-CHIP version. This version consolidated the CPU and the two PPUs (Picture Processing Units) into a single ASIC (Application-Specific Integrated Circuit), reducing the total number of chips on the motherboard to nine.

📌This reduction not only simplified the design but also potentially improved the system's reliability and performance.

Motherboard Versions

📌Over its 12-year lifespan, Nintendo released twelve different versions of the SNES motherboard.

📌These versions include various models like SHVC-CPU-01, SNS-CPU-GPM-01, and SNS-CPU-1CHIP-01 among others, each corresponding to different production years and design tweaks.

📌The versions are categorized into four major generations: Classic, APU, 1-CHIP, and Junior, with the 1-CHIP and Junior versions representing the most significant redesigns.

📌The Super Nintendo Jr (also known as Mini) is noted as the final form of the SNES, maintaining the reduced chip count and featuring a more integrated design where the motherboard no longer has parts dedicated to specific subsystems.

Evolution of the SNES Motherboard:

Over its 12-year lifespan, Nintendo released twelve versions of the SNES motherboard, each reducing the number of chips and components. The most notable advancement was the 1-CHIP version, which integrated the CPU and two PPUs into a single ASIC, simplifying the design and potentially enhancing performance. It sheds light on the technical marvels and challenges of the SNES cartridge system, highlighting how Nintendo leveraged additional hardware within cartridges to push the boundaries of what was possible in video gaming during the era

Enhancement Processors

📌SNES cartridges were notable for their ability to include more than just game instructions and assets. They could also house additional hardware components such as the CIC copy protection chip, SRAM, and enhancement processors.

📌These enhancement processors, such as the "Super Accelerator 1" (SA-1) chip, significantly boosted the SNES's capabilities. The SA-1 chip, found in 34 cartridges, was a 65C816 CPU running at 10.74 MHz—four times faster than the SNES's main CPU. It also included 2KiB of SRAM and an integrated CIC.

Copy-Protection Mechanism

📌The SNES utilized a copy-protection mechanism involving two CIC chips that communicated in lockstep—one in the console and the other in the cartridge. If the console's CIC detected an unauthorized game, it would reset every processor in the system.

📌Some unsanctioned games, like "Super 3D Noah's Ark," bypassed this protection by requiring an official cartridge to be plugged on top of them, using the official game's CIC to authenticate.

Game Enhancements

📌The inclusion of enhancement processors allowed for significant improvements in game performance and graphics. For example, the SA-1 chip enabled the SNES to animate and detect collisions on all 128 sprites available in the PPU, transform sprites on the fly (rotate/scale), and write them back into the PPU VRAM.

📌Another enhancement chip, the Super-GFX, excelled at rendering pixels and rasterizing polygons, usually rendering into a framebuffer located on the cartridge. This content was then transferred to the VRAM during VSYNC.

Regional Compatibility and Circumvention

📌The article also touches on the physical and electronic measures Nintendo used to enforce regional compatibility, such as the different shapes of cartridges and the CIC lockout system. However, it mentions that these measures were not foolproof and could be circumvented.

Community and Development Insights

📌Discussions on platforms like Hacker News reflect on the impact and potential of these cartridges, comparing them to other Nintendo innovations and discussing the technical challenges and solutions provided by the SNES's design

Clock Generators in the SNES**:**

The SNES utilized two main clock generators to manage the timing for its various components. These clocks were crucial for the operation of the CPU, PPU, and APU. The system also included enhancement chips in some cartridges, which used these clocks for additional processing power, exemplified by the SuperFX chip used in games like StarFox. This detailed examination of the SNES's clock system reveals the intricate design and engineering that supported the console's complex graphics and audio capabilities, allowing for advanced gaming experiences during its era.

Clock Generators

📌The SNES motherboard features two primary clock generators located in the X2 and X1 slots.

📌The X2 slot houses a 24.576 MHz ceramic resonator, which is blue in color. This resonator is crucial for the operation of the Audio Processing Unit (APU), setting the pace for audio processing on the SNES.

📌The X1 slot contains a 21.300 MHz oscillator, labeled D21L3, which is yellow. This oscillator is strategically placed near the CPU and the Picture Processing Unit (PPU), thereby setting their operational pace.

Clock Distribution and Enhancement Chips

📌The SNES utilizes these master clocks in conjunction with dividers to generate additional clocks needed by various components. For instance, the Ricoh 5A22 CPU operates at 1/6th the frequency of the master clock, resulting in a frequency of 3.579545 MHz.

📌The system includes a total of fifteen different clocks, highlighting the complex timing management within the SNES.

📌The SYS-CLK line, which runs at 21.47727 MHz, is routed to the cartridge port. This setup is not typically necessary for the basic operation of the cartridges, which contain ROM with game data and instructions. However, this clock signal is crucial for cartridges that contain their own enhancement processors, like the SuperFX chip used in games such as StarFox.

📌These enhancement chips can utilize the SYS-CLK for additional processing power, with some chips like the MARIO version of the SuperFX processor using an internal divider to adjust the clock frequency to suit specific processing needs.

Impact on Game Performance

📌The precision of these clock generators is vital for the deterministic execution of game code, which is particularly important for applications like tool-assisted speedruns (TAS). Over time, the accuracy of ceramic resonators can degrade, leading to performance inconsistencies

Architecture of Consoles: A Practical Analysis

Rodrigo Copetti’s series of books, “Architecture of Consoles: A Practical Analysis, “ dives deep into the fascinating world of video game consoles, uncovering the secrets behind their mind-boggling technology. But let’s be honest, who needs a social life when you can spend your time dissecting the inner workings of these magical boxes, right?

In this series, the author takes us on a wild ride through the evolution of consoles, proving that they’re more than just a bunch of numbers and fancy jargon. From the Nintendo 3DS to the Xbox and PlayStation series, these books show that consoles are like snowflakes — each one is unique and special in its own way.

So, if you’re ready to trade your social life for a deep dive into the mesmerizing world of console architecture, Copetti’s books are just the ticket. They’re a treasure trove of technical knowledge, perfect for anyone who’s ever wondered what makes these magical boxes tick.

These books are part of a series on console architecture, and it is structured similarly to his previous work on the PS3's architecture. This allows readers who are familiar with the PS3's architecture to compare the two consoles side-by-side. Books on console architecture, including “PlayStation 3 Architecture”, are targeted towards individuals with a basic knowledge of computing who are interested in the evolution and internal workings of video game consoles. His writings are not developer manuals but rather in-depth introductions to how each system works internally. He tries to adapt his content for wider audiences, so even those without a deep understanding of computing can still find value in his work. His books are appreciated by both technical and non-technical readers for their in-depth yet accessible explanations of complex console architectures. Therefore, his target audience can be considered quite broad, encompassing anyone from casual readers with an interest in technology to professionals in the gaming industry, computer engineers, and enthusiasts of console gaming and hardware.

Some other books by this author

📌NES Architecture: More than a 6502 machine

📌Game Boy Architecture

📌Super Nintendo Architecture

📌PlayStation Architecture

📌Nintendo 64 Architecture

📌GameCube Architecture

📌Wii Architecture

📌Nintendo DS Architecture

📌Master System Architecture

Xbox Original

If you are not familiar with Xbox original, it’s suggested to start with reading Xbox Arch before Xbox 360. “Xbox Architecture” The book provides an in-depth look at the console’s architecture, focusing on its unique features and the technological innovations that set it apart from its competitors. The book begins by discussing the historical context of the Xbox’s development, noting that Microsoft aimed to create a system that would be appreciated by developers and welcomed by users due to its familiarities and online services.

📌**One of the main topics covered in the book is the Xbox’s CPU.** The console uses a slightly customized version of the Intel Pentium III, a popular off-the-shelf CPU for computers at the time, running at 733 MHz. The book explores the implications of this choice and how it contributes to the overall architecture of the Xbox.

📌**The book also delves into the Graphics of the Xbox.** It uses a custom implementation of Direct3D 8.0, which was extended to include Xbox-specific features. This allowed PC developers to port their games to the Xbox with minimal changes

📌**The Development Ecosystem of the Xbox is another key topic covered in the book.** Game development on the Xbox is complex, with various libraries and frameworks interacting with the console’s hardware. The book provides a detailed analysis of this ecosystem, helping readers understand the intricacies of game development on the Xbox

📌**The Network Service of the Xbox is also discussed.** The Xbox included an Ethernet connection and a centralized online infrastructure called Xbox Live, which were innovative features at the time. The book explores how these features contribute to the overall architecture of the Xbox

📌**Finally, the book also covers the Security aspects of the Xbox, including its anti-piracy system.** It explains how this system works and how it fits into the console’s overall architecture

Xbox Original Architecture quick facts

📌The original Xbox used a familiar system for developers and online services for users

📌The Xbox CPU is based on Intel’s Pentium III with the P6 microarchitecture

📌The console has 64 MiB of DDR SDRAM, which is shared across all components

📌The Xbox GPU is manufactured by Nvidia and is called the NV2A

📌The original Xbox controller, called The Duke, was replaced with a new revision called Controller S due to criticism

Xbox 360

The book “Xbox 360 Architecture: A Supercomputer for the Rest of Us” provides an in-depth analysis of the Xbox 360's architecture, discussing its design, capabilities, and the technological innovations it introduced and, explaining how the console works internally. It is a valuable resource for anyone interested in the evolution of gaming console technology. The book is part of the “Architecture of Consoles: A Practical Analysis” series, which looks at the evolution of video game consoles and their unique ways of working.

The book begins with a brief history of the Xbox 360, which was released a year before its main competitor, the PlayStation 3. It discusses the business aspect of the Xbox 360's CPU and the sequence of events that led to its development.

The book then delves into the technical aspects of the Xbox 360's architecture. It discusses the console’s CPU, which was a significant departure from the single-core CPU used in the original Xbox. The Xbox 360's CPU, known as Xenon, was a triple-core processor designed by IBM. Each core was capable of handling two threads simultaneously, allowing up to six threads to be processed at once.

The book also discusses the Xbox 360's GPU, known as Xenos, which was designed and manufactured by ATI. The GPU was based on a new architecture and could deliver 240 GFLOPS of performance. The Xenos GPU introduced the concept of a unified shader pipeline, which combined two different dedicated pipelines for increased performance.

The book further discusses the Xbox 360's main memory, which was a significant increase over the original Xbox’s 64 MB. This allowed for more complex games and applications to be run on the console.

The book also covers the Xbox 360's operating system, development ecosystem, and network service. It discusses how the console’s architecture was designed to be flexible and easy to program for, with a balanced hardware architecture that could adapt to different game genres and developer needs.

The main topics covered in the book include:

📌**CPU:** The book delves into the details of the Xbox’s CPU, discussing its unique features and how it compares to the CPUs of other consoles. It also provides a historical context, explaining how the CPU’s design was influenced by the technological trends and challenges of the time

📌**Graphics:** The book provides a detailed analysis of the Xbox’s graphics capabilities, including its use of a semi-customised version of Direct3D 9 and how this influenced future revisions of Direct3D

📌**Security:** The book discusses the Xbox’s anti-piracy system, explaining how it works and how it contributes to the console’s overall architecture

📌**Development Ecosystem:** The book explores the complexities of game development on the Xbox, discussing the various libraries and frameworks used and how they interact with the console’s hardware

📌**Network Service:** The book also covers the Xbox’s online capabilities, discussing its Ethernet connection and the Xbox Live online infrastructure

Xbox 360 Architecture quick facts

📌The Xbox 360 was released a year before its main competitor, the PS3

📌The Xbox 360's CPU, called Xenon, is a multi-core processor developed by IBM

📌The console uses a semi-customized version of Direct3D 9 for its GPU, called Xenos

📌The Xbox 360 has a unified memory architecture with 512 MB of GDDR3 RAM

PS2

“PlayStation 2 Architecture” provides an in-depth analysis of the PlayStation 2 console’s internal workings. Despite not being the most powerful console of its generation, the PlayStation 2 achieved a level of popularity that was unthinkable for other companies. The book explains that the PlayStation 2's success was due to its Emotion Engine, a powerful package designed by Sony that ran at ~294.91 MHz. This chipset contained multiple components, including the main CPU and other components designed to speed up certain tasks. The book also discusses the PlayStation 2's operating system, which relied on the Image Processing Unit (IPU) for DVD playback and compressed High-resolution textures. The PlayStation 2's development ecosystem is also covered, with Sony providing the hardware and software to assist game development

PS2 Architecture quick facts

📌The PlayStation 2 (PS2) was not the most powerful console of its generation but achieved immense popularity

📌The Emotion Engine (EE) is the heart of the PS2, running at ~294.91 MHz and containing multiple components, including the main CPU

📌The main core is a MIPS R5900-compatible CPU with various enhancements

📌The PS2 uses Vector Processing Units (VPUs) to enhance its processing capabilities

📌The console has backward compatibility with the original PlayStation through the use of an I/O Processor (IOP)

📌The PS2 introduced the DualShock 2 controller, which featured two analog sticks and two vibration motors

📌The operating system of the PS2 is stored on a 4 MB ROM chip

PS3

“PlayStation 3 Architecture” offers a comprehensive analysis of the PlayStation 3 console’s internal structure. The book explains that the PlayStation 3's underlying hardware architecture continues the teachings of the Emotion Engine, focusing on vector processing to achieve power, even at the cost of complexity. The PlayStation 3's CPU, the Cell Broadband Engine, is a product of a crisis of innovation and had to keep up as trends for multimedia services evolved. The book also discusses the PlayStation 3's main memory and the Synergistic Processor Element (SPE), which are accelerators included within the PS3's Cell. The PlayStation 3 also contains a GPU chip manufactured by Nvidia, called Reality Synthesizer or 'RSX', which runs at 500 MHz and is designed to offload part of the graphics pipeline

PS3 Architecture quick facts

📌The PS3 focuses on vector processing to achieve power, even at the cost of complexity

📌The Cell Broadband Engine is the main processor of the PS3, developed jointly by Sony, IBM, and Toshiba

📌The PS3's CPU is massively complex and features a Power Processing Element (PPE) and multiple Synergistic Processor Elements (SPEs)

📌The PS3 uses a GPU chip called Reality Synthesizer (RSX) manufactured by Nvidia

There are several notable differences in architectures are discussed in the books

Xbox 360 and Xbox Original

📌**CPU:** The original Xbox relied on popular off-the-shelf stock (Intel's Pentium III) with slight customizations. This was a single-core CPU extended with vectorized instructions and a sophisticated cache design. On the other hand, the Xbox 360 introduced a new type of CPU that was unlike anything seen on the store shelves. This was a multi-core processor developed by IBM, reflecting an obsessive need for innovation characteristic of the 7th generation of consoles

📌**GPU:** The original Xbox's GPU was based on the NV20 architecture, with some modifications to work in a unified memory architecture (UMA) environment. The Xbox 360, however, used a semi-customized version of Direct3D 9 for its GPU, called Xenos

📌**Memory:** The original Xbox included a total of 64 MiB of DDR SDRAM, which was shared across all components of the system. The Xbox 360, on the other hand, had a unified memory architecture with 512 MB of GDDR3 RAM

📌**Development Ecosystem:** The original Xbox was designed with familiarities appreciated by developers and online services welcomed by users. The Xbox 360, however, was designed with an emphasis on the emerging 'multi-core' processor and unorthodox symbiosis between components, which enabled engineers to tackle unsolvable challenges with cost-effective solutions

📌**Release Timing:** The Xbox 360 was released a year before its main competitor, the PlayStation 3, and was already claiming technological superiority against the yet-to-be-seen PlayStation 3

PS2 and PS3:

📌**CPU:** The PS2's Emotion Engine was designed by Toshiba, using MIPS technology, and focused on achieving acceptable 3D performance at a reduced cost. In contrast, the PS3's CPU, the Cell Broadband Engine, was developed through a collaboration between Sony, IBM, and Toshiba, and is a highly complex and innovative processor that intersects complex needs and unusual solutions

📌**GPU:** The PS2's GPU, the Graphics Synthesizer, was a fixed-functionality GPU designed for 3D performance. The PS3's GPU, the Reality Synthesizer (RSX), was manufactured by Nvidia and was designed to offload part of the graphics pipeline, offering better parallel processing capabilities

📌**Memory:** The PS2 had 32 MB of RDRAM, while the PS3 had a more advanced memory system, with 256 MB of XDR DRAM for the CPU and 256 MB of GDDR3 RAM for the GPU.

📌**Development Ecosystem:** The PS2's development ecosystem was based on MIPS technology and focused on achieving acceptable 3D performance at a reduced cost. The PS3's development ecosystem was more complex, involving collaboration between Sony, IBM, and Toshiba, and focused on creating a powerful and innovative system

📌**Backward Compatibility:** The PS2 was backward compatible with PS1 games through the inclusion of the original PS1 CPU and additional hardware components. The PS3 also offered backward compatibility with PS2 games, but this was achieved through software emulation in later revisions of the console

PS2 and Xbox Original:

📌**CPU:** The PS2's Emotion Engine was designed by Toshiba, using MIPS technology, and focused on achieving acceptable 3D performance at a reduced cost. In contrast, the Xbox Original's CPU was based on Intel's Pentium III, which was a popular off-the-shelf stock with slight customizations

📌**GPU:** The PS2's GPU, the Graphics Synthesizer, was a fixed-functionality GPU designed for 3D performance. The Xbox Original's GPU was based on the NV20 architecture, with some modifications to work in a unified memory architecture (UMA) environment

📌**Memory:** The PS2 had 32 MB of RDRAM, while the Xbox Original included a total of 64 MiB of DDR SDRAM, which was shared across all components of the system

📌**Development Ecosystem:** The PS2's development ecosystem was based on MIPS technology and focused on achieving acceptable 3D performance at a reduced cost. The Xbox Original was designed with familiarities appreciated by developers and online services welcomed by users

PS3 and Xbox 360:

📌**CPU:** The PS3's CPU, the Cell Broadband Engine, is a highly complex and innovative processor that intersects complex needs and unusual solutions. It was developed through a collaboration between Sony, IBM, and Toshiba. On the other hand, the Xbox 360's CPU, Xenon, was a new type of CPU that was unlike anything seen on the store shelves. It reflects an obsessive need for innovation, a peculiar trait of that era

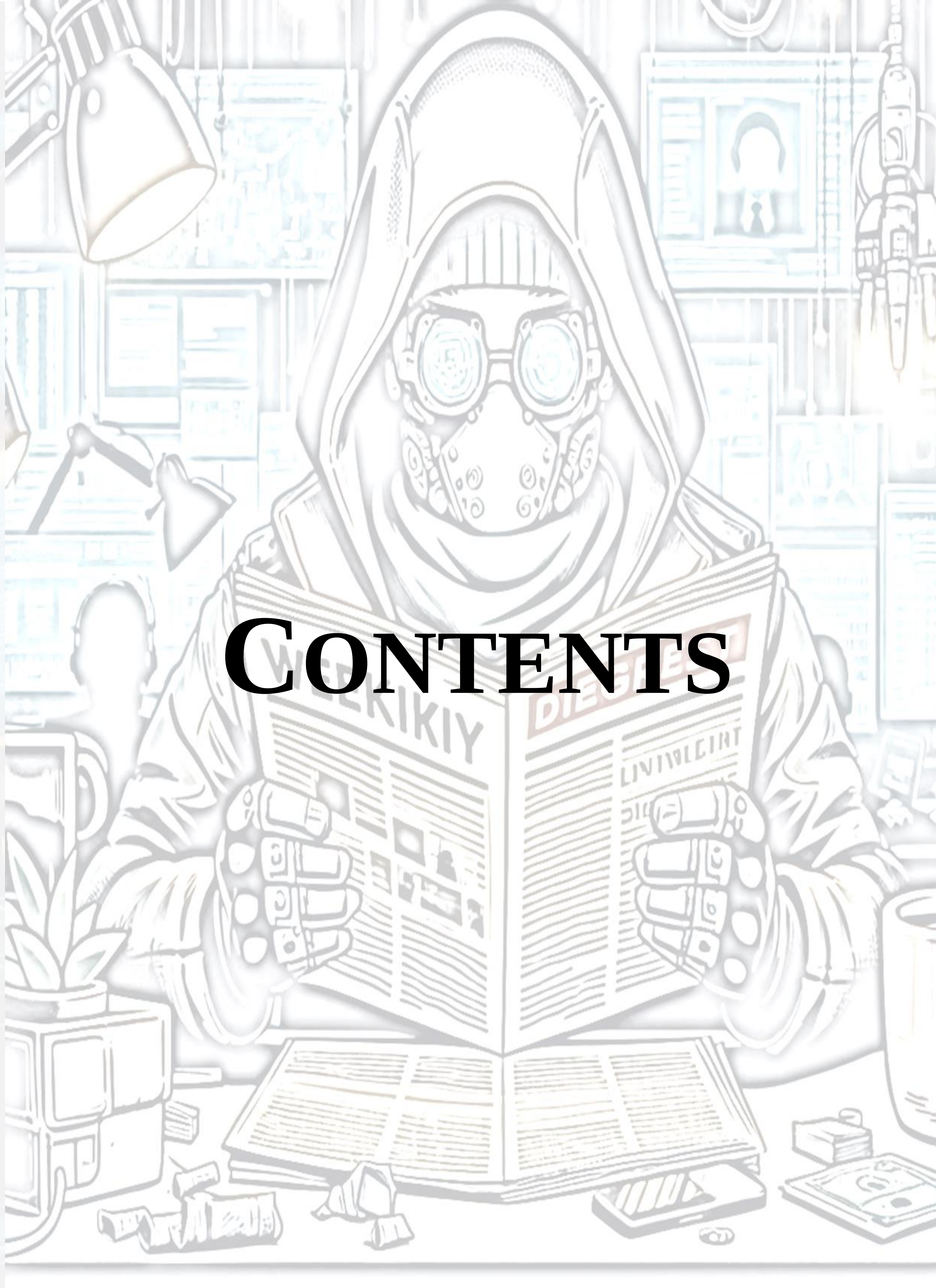
📌**GPU:** The PS3's GPU, the Reality Synthesizer or 'RSX', was manufactured by Nvidia and was designed to offload part of the graphics pipeline. The Xbox 360's GPU, Xenos, was a semi-customised version of Direct3D 9 that makes room for the extra functions of Xenos

📌**Memory:** The PS3's memory was distributed across different memory chips, and while it didn't implement a UMA architecture, it could still distribute graphics data across different memory chips if programmers decide to do so.

📌**Development Ecosystem:** The PS3's development ecosystem was based on the Cell Broadband Engine, a joint project between Sony, IBM, Toshiba, and Nvidia. The Xbox 360's development ecosystem was based on the Xenon CPU and the semi-customized version of Direct3D 9

DIGREST

CONTENTS



Read more: Boosty | Sponsr | TG

Left over Locals

In a twist of snarky irony, the very technology that powers our AI and machine learning models is now the target of a new vulnerability, dubbed "LeftoverLocals". Disclosed by Trail of Bits, this security flaw allows the recovery of data from GPU local memory created by another process, affecting Apple, Qualcomm, AMD, and Imagination GPUs. In this document, we provide a detailed analysis of the "LeftoverLocals" CVE-2023-4969 vulnerability, which has significant implications for the integrity of GPU applications, particularly for large language models (LLMs) and machine learning (ML) models executed on affected GPU platforms, including those from Apple, Qualcomm, AMD, and Imagination. This document provides valuable insights for cybersecurity professionals, DevOps teams, IT specialists, and stakeholders in various industries. The analysis is designed to enhance the understanding of GPU security challenges and to assist in the development of effective strategies to safeguard sensitive data against similar threats in the future.

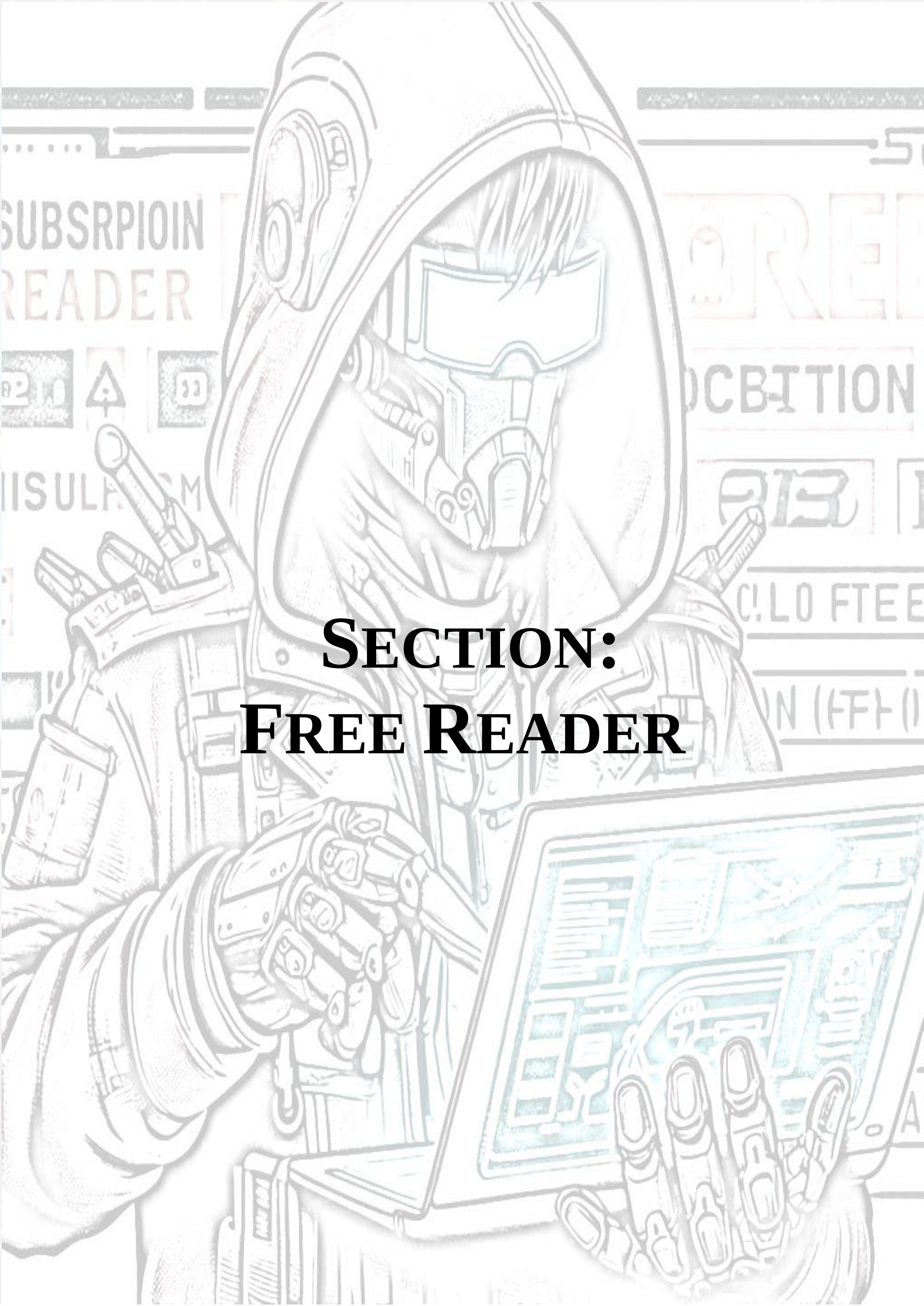
PulseVPN Vulnerability / CVE-2023-38043, CVE-2023-35080, CVE-2023-38543

PureVPN presents itself as a beacon of online privacy and security in the vast and murky waters of the internet.In the grand tradition of "security first", we find ourselves marveling at the latest contributions to the cybersecurity hall of fame: CVE-2023-38043, CVE-2023-35080, and CVE-2023-38543. These vulnerabilities, discovered in the Avanti Secure Access Client, previously known as Pulse Secure VPN, have opened up a new chapter in the saga of "How Not To VPN".

This analysis is particularly beneficial for security professionals seeking to understand the intricacies of VPN vulnerabilities and their implications for enterprise security. It also serves as a resource for IT administrators responsible for maintaining secure VPN configurations and for industry stakeholders interested in the broader implications of such vulnerabilities on digital security and compliance.

Read more: Boosty | Sponsr | TG

Section: Free Reader



Left Over Locals

Abstract – In this document, we provide a detailed analysis of the "LeftoverLocals" CVE-2023-4969 vulnerability, which has significant implications for the integrity of GPU applications, particularly for large language models (LLMs) and machine learning (ML) models executed on affected GPU platforms, including those from Apple, Qualcomm, AMD, and Imagination.

This document provides valuable insights for cybersecurity professionals, DevOps teams, IT specialists, and stakeholders in various industries. The analysis is designed to enhance the understanding of GPU security challenges and to assist in the development of effective strategies to safeguard sensitive data against similar threats in the future.

A. Introduction

Trail of Bits has disclosed a vulnerability named LeftoverLocals, which allows the recovery of data from GPU local memory that was created by another process. This vulnerability affects Apple, Qualcomm, AMD, and Imagination GPUs and has significant implications for the security of GPU applications, especially large language models (LLMs) and machine learning (ML) models run on the affected platforms.

The vulnerability enables an attacker to listen in on another user's interactive LLM session across process or container boundaries. Moreover, the vulnerability is significant in the context of LLMs and ML models because it can lead to the leakage of sensitive data involved in training these models.

B. Vulnerable environments

The LeftoverLocals vulnerability can be exploited in various environments, including cloud providers, mobile applications, and potentially even in remote attacks.

- **Cloud Providers**: Cloud providers often offer GPU resources to their customers, which are shared among multiple users. In such multi-tenant environments, the LeftoverLocals vulnerability can be exploited if an attacker has access to the same physical GPU as the victim. This could allow the attacker to recover data from the GPU's local memory that was created by another process, leading to significant data leakage. This is particularly concerning for applications that use large language models (LLMs) and machine learning (ML) models, as these applications often handle sensitive data.
- **Mobile Applications**: Mobile devices that use vulnerable GPUs are also at risk. For example, Apple has acknowledged that devices such as the iPhone 12 and M2 MacBook Air are affected by the LeftoverLocals vulnerability.
- **Remote Attacks**: LeftoverLocals vulnerability could potentially be exploited remotely in scenarios where an attacker has compromised a system and gained the ability to run custom code, or in environments where users are allowed to run custom GPU compute applications.

C. Leftoverlocals vs. other vulnerabilities

1) Leftoverlocals vs. GPU vulnerabilities

The LeftoverLocals vulnerability is distinct from other GPU vulnerabilities primarily in its method of data leakage through GPU local memory. Unlike many vulnerabilities that exploit specific software bugs or hardware flaws, LeftoverLocals is based on the failure of GPU frameworks to completely isolate memory between processes. This allows an adversary to run a GPU compute application to read data left in the GPU local memory by another user.

Other GPU vulnerabilities might target different aspects of GPU architecture or software, such as buffer overflows, race conditions, or driver-level exploits. These vulnerabilities often require specific conditions to be met or rely on complex interactions between software and hardware.

The leaked data can be substantial enough to reconstruct the models or responses, posing a significant risk to the confidentiality of the processed information.

The severity of the LeftoverLocals vulnerability is high due to several factors:

- **Broad Impact**: The vulnerability affects a wide range of GPUs from major manufacturers like AMD, Apple, Qualcomm, and Imagination Technologies.
- **Data Leakage**: LeftoverLocals can leak significant amounts of data. For instance, on an AMD Radeon RX 7900 XT GPU, it can leak about 5.5 MB of data per GPU invocation, which can amount to about 181 MB for each LLM query. This is sufficient to reconstruct the LLM response with high precision.
- **Ease of Exploitation**: The vulnerability can be exploited by simply running a GPU compute application to read data left in the GPU local memory by another user.
- **Mitigation Challenges**: Mitigating the vulnerability may be difficult for many users. One suggested mitigation is modifying the source code of all GPU

kernels that use local memory to store 0 to any local memory locations that were used in the kernel before it ends. However, this might impact performance.

- **Sensitive Data Exposure**: The vulnerability is particularly concerning in the context of AI and machine learning, where sensitive data is often used in training models.

2)Leftoverlocals vs. CPU vulnerabilities

Spectre and Meltdown are CPU vulnerabilities that exploit "side-channel" attacks, which involve extracting information from the physical implementation of computer systems rather than software bugs or errors.

Spectre tricks other applications into accessing arbitrary locations in their memory. Meltdown, on the other hand, breaks the fundamental isolation between user applications and the operating system, allowing an application to access all system memory, including memory allocated for the kernel.

In terms of severity, all three vulnerabilities are serious as they can lead to unauthorized access to sensitive data. However, they differ in their scope and the nature of the data they can expose. LeftoverLocals primarily affects GPU applications and can lead to the leakage of data from LLMs and ML models. Spectre and Meltdown, on the other hand, can potentially expose any data processed by the CPU, including passwords, encryption keys, and other sensitive information.

The potential consequences of these vulnerabilities are severe:

- They affect almost all CPUs released since 1995, making their impact extremely widespread.
- They can potentially expose any data processed by the CPU, including passwords, encryption keys, and other sensitive information.
- They are hard to detect as the exploitation does not leave any traces in traditional log files.

3)Similarities

The vulnerabilities differ in their specific mechanisms and the domains they affect (GPUs for LeftoverLocals and CPUs for Spectre/Meltdown). Spectre and Meltdown are also considered to be more pervasive and difficult to mitigate due to their presence in CPUs used in a vast array of devices over the past couple of decades.

The LeftoverLocals vulnerability shares some similarities with the Spectre and Meltdown vulnerabilities in terms of their implications for security:

- **Data Leakage**: Both LeftoverLocals and Spectre/Meltdown allow unauthorized access to sensitive data. LeftoverLocals enables data recovery from GPU local memory, while Spectre and Meltdown exploit CPU speculative execution to access protected memory.
- **Exploitation of Hardware Features**: Both sets of vulnerabilities exploit hardware features designed for performance optimization—GPU local memory in the case of LeftoverLocals, and speculative execution in CPUs for Spectre and Meltdown.
- **Cross-Process Boundary Violation**: They both violate process isolation guarantees. LeftoverLocals reads data across process or container boundaries on GPUs, and Spectre/Meltdown can read data across application boundaries on CPUs.
- **Affecting Multiple Vendors**: Both vulnerabilities impact products from multiple vendors. LeftoverLocals affects GPUs from Apple, Qualcomm, AMD, and Imagination Technologies, while Spectre and Meltdown affect CPUs from Intel, AMD, and ARM.
- **Complex Mitigation**: Mitigating both vulnerabilities is non-trivial. LeftoverLocals may require changes to GPU kernel code, while Spectre and Meltdown have required a combination of microcode updates, operating system patches, and in some cases, hardware redesigns.
- **Stealthy Nature of Attacks**: Attacks exploiting these vulnerabilities are difficult to detect as they do not leave traditional traces in log files, making it challenging to determine if they have been used in real-world attacks.

D. LeftOverLocal Exploitation requirements

- **Shared Access to a GPU**: An attacker needs shared access to a GPU via a programmable interface.
- **Listener-Writer Model**: The exploitation process involves two different programs: a Listener and a Writer. The Writer stores specific values (referred to as "canary values") in local memory, while the Listener reads uninitialized local memory to check for these canary values.
- **Access to Devices**: The attacker needs access to the devices.

1)Shared Access to a GPU

The exploitation of the LeftoverLocals vulnerability requires shared access to a GPU, which is a common scenario in multi-tenant environments where multiple users or applications may be utilizing the same physical GPU resources. This can occur in cloud computing platforms, shared data centers, or any system where GPU resources are allocated dynamically to different users or tasks. In such environments, the GPU's local memory is not always properly cleared between different kernel executions or between the usages by different processes. This oversight allows for the possibility that sensitive data from one process could be left in the local memory and subsequently accessed by another process that is scheduled to use the same GPU.

2)Listener-Writer Model

The Listener-Writer model is a method used to exploit the LeftoverLocals vulnerability. It involves two different programs: a Listener and a Writer. These programs interact with the GPU's local memory to demonstrate the vulnerability.

The Writer program is designed to intentionally store specific values, referred to as "canary values," in the GPU's local memory. These values are unique and identifiable, serving as markers that can be detected later. The Writer program does not clear these values from the local memory after it finishes its execution.

The Listener program is designed to read uninitialized local memory on the GPU. It scans the local memory looking for the canary values that the Writer program left behind. If the Listener program detects these canary values, it indicates that the local memory was not properly cleared between the execution of different programs.

3) Access to Devices

Access to devices is a critical requirement for exploiting the LeftoverLocals vulnerability. Attackers need to have some level of operating system access on the target device to exploit the vulnerability. This access doesn't necessarily need to be root or administrative access; it could be any level of access that allows the attacker to execute their own GPU compute applications.

In the case of Apple devices, the company has acknowledged that devices such as the iPhone 12 and M2 MacBook Air are affected by the LeftoverLocals vulnerability. While Apple has shipped fixes with its latest hardware, millions of existing devices that rely on previous generations of Apple silicon remain potentially vulnerable.

Qualcomm and AMD have also confirmed the impact of the vulnerability on their GPUs and have taken steps to address it. Qualcomm has released firmware patches, and AMD has detailed plans to offer optional mitigations should be released

E. Proccess flow & PoC

The LeftoverLocals vulnerability can be exploited using a method that involves modification, fingerprinting the model, and listening to the LLM output.

1) Modification

The first step in exploiting the LeftoverLocals vulnerability involves modifying the GPU kernel code. The researchers at Trail of Bits were able to modify the GPU kernel code to read and write to the GPU's local memory. This allowed them to create a proof-of-concept (PoC) where an attacker can listen into another user's interactive LLM session across process or container boundaries.

2) Fingerprinting the Model

Fingerprinting the model involves identifying the specific LLM being used. This can be done by observing the GPU memory usage patterns of the LLM. Different LLMs will have different memory usage patterns, and by observing these patterns, an attacker can determine which LLM is being used. This information can be used to tailor the attack to the specific LLM, increasing the chances of successfully exploiting the vulnerability.

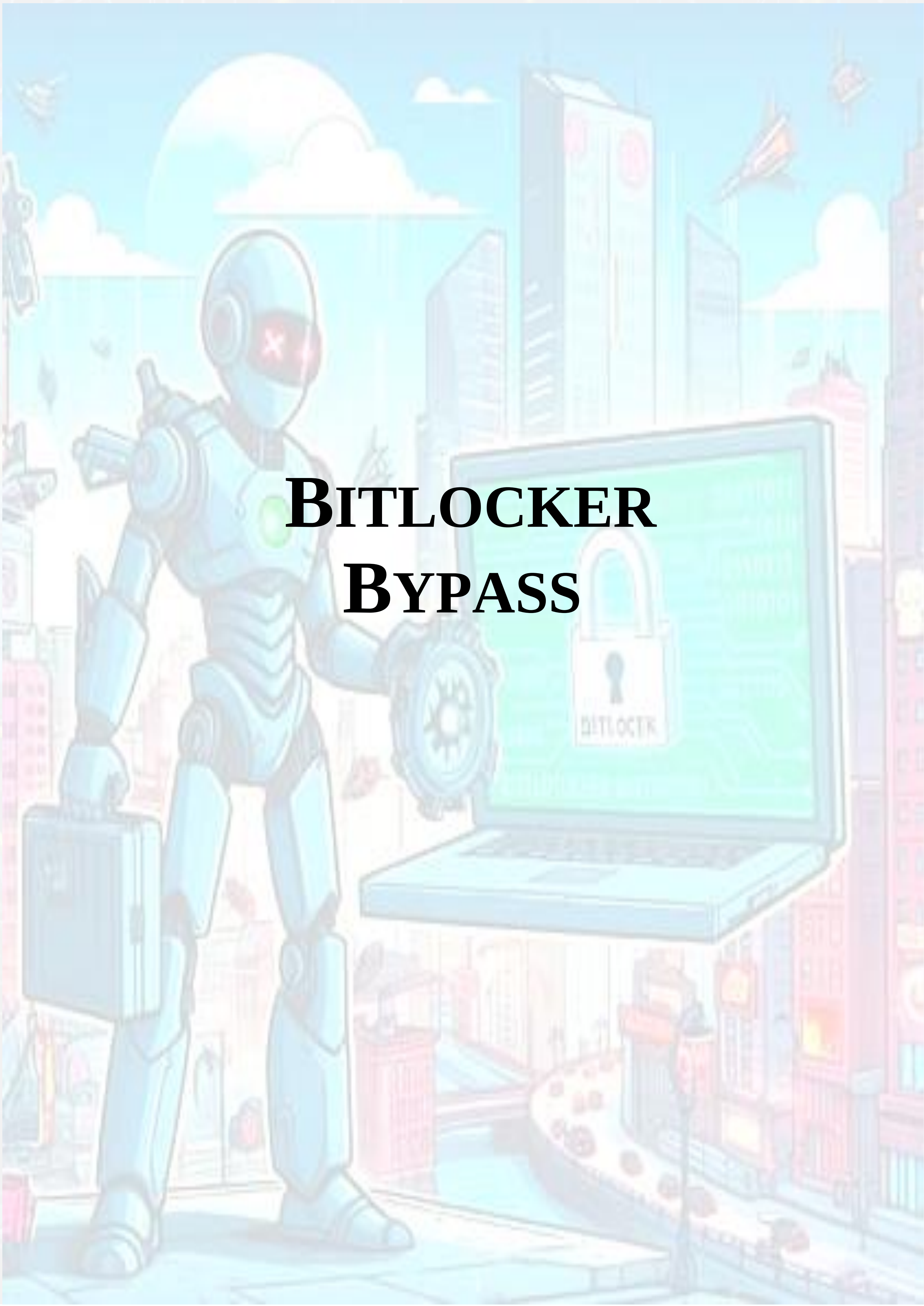
3) Listening to the LLM Output

Once the model has been fingerprinted, the attacker can then start listening to the LLM output. This involves repeatedly launching a GPU kernel that reads from uninitialized local memory on the GPU. The attacker scans the local memory looking for specific values that the LLM has left behind. If these values are detected, it indicates that the local memory was not properly cleared between the execution of different programs. This allows the attacker to recover data from the LLM's computations, leading to significant data leakage.

4) PoC

The proof-of-concept (PoC) was developed using OpenCL, a framework for writing programs that execute across heterogeneous platforms and built by the researchers at Trail of Bits to demonstrate the LeftoverLocals vulnerability has several key features:

- **Model Fingerprinting**: The PoC involves identifying the specific large language model (LLM) being used by observing the GPU memory usage patterns. Different LLMs have different memory usage patterns, which can be used to determine which LLM is being used.
- **Listening to LLM Output**: The PoC involves repeatedly launching a GPU kernel that reads from uninitialized local memory on the GPU. The attacker scans the local memory looking for specific values that the LLM has left behind. If these values are detected, it indicates that the local memory was not properly cleared between the execution of different programs, allowing the attacker to recover data from the LLM's computations.
- **Data Leakage**: The researchers found that the LeftoverLocals vulnerability can leak approximately 5.5 MB per GPU invocation on an AMD Radeon RX 7900 XT, which, when running a 7B model, adds up to about 181 MB for each LLM query. This is enough information to reconstruct the LLM response with high precision.
- **Cross-Process or Container Boundaries**: The PoC demonstrates that an attacker can listen into another user's interactive LLM session across process or container boundaries. This shows that the vulnerability can be exploited in multi-tenant environments, such as cloud computing platforms, where multiple users share the same physical GPU.
- **Access to Devices**: The PoC requires the attacker to have access to the target device. This could be any level of access that allows the attacker to execute their own GPU compute applications.



Bitlocker Bypass

Abstract – This document provides a comprehensive analysis of the method demonstrated in the video "Breaking Bitlocker - Bypassing the Windows Disk Encryption" where the author showcases a low-cost hardware attack capable of bypassing BitLocker encryption. The analysis will cover various aspects of the attack, including the technical approach, the use of a Trusted Platform Module (TPM) chip, and the implications for security practices.

The analysis provides a high-quality summary of the demonstrated attack, ensuring that security professionals and specialists from different fields can understand the potential risks and necessary countermeasures. The document is particularly useful for cybersecurity experts, IT professionals, and organizations that rely on BitLocker for data protection and to highlight the need for ongoing security assessments and the potential for similar vulnerabilities in other encryption systems.

A. *Introduction*

In the video "Breaking Bitlocker - Bypassing the Windows Disk Encryption", the author is talking about a method to bypass the Windows Disk Encryption (BitLocker) using different attacks including using a low-cost hardware attack. He shows how an attacker can use a simple device to extract the encryption key from a computer's TPM (Trusted Platform Module) chip, which is used to store the encryption key for BitLocker. This attack allows the attacker to decrypt the computer's hard drive and access the data without knowing the BitLocker password.

The video provides:

- The method to bypass BitLocker using a low-cost hardware attack.
- The attack targets the TPM chip, which is used to store the encryption key for BitLocker.
- The detailed explanation of the attack, including the hardware and software components involved.
- The implications of this attack and provides recommendations for how users can protect their data from this type of attack.

B. *Methodology*

The methodology for analyzing BitLocker involves several steps:

- **Understanding the Technical Details**: it begins by thoroughly understanding the technical aspects of BitLocker, including its encryption algorithms, key management mechanisms, and security features. This knowledge is essential for identifying potential vulnerabilities and weaknesses in the system.
- **TPM Bypass Attack Demonstration**: it provides a detailed explanation of the TPM bypass attack, including the hardware and software components required to provide strong visual evidence of attack in practice, showing how an attacker can extract the encryption key from a computer's TPM chip using a simple device.
- **Analysis of BitLocker's Encryption Algorithms**: it analyzes BitLocker's encryption algorithms, including AES and XTS-AES, and discusses their strengths and weaknesses. It also examines the key management mechanisms used by BitLocker and how they can be exploited by attackers. This analysis provides a deeper understanding of the vulnerabilities in BitLocker and helps viewers appreciate the significance of the attack.
- **Vulnerability Analysis**: Based on the technical understanding, literature review, and practical testing, it performs a comprehensive vulnerability analysis of BitLocker. This involves identifying potential attack vectors, exploiting vulnerabilities, and assessing the impact of these vulnerabilities on the security of BitLocker.
- **Practical Testing and Experimentation**: It conducts practical tests and experiments to evaluate the effectiveness of BitLocker's security features. This may involve setting up test environments, simulating attacks, and analyzing the results to identify potential weaknesses.
- **Developing Countermeasures and Recommendations**: Finally, he develops countermeasures and recommendations to mitigate the identified vulnerabilities and improve the overall security of BitLocker. These recommendations may include configuration best practices, security updates, and additional security measures to enhance the protection of data encrypted with BitLocker.

C. *Security weaknesses viewpoint*

The attack is possible due to several factors:

- **Weak Encryption Algorithms**: BitLocker uses weak encryption algorithms, such as AES-128 and XTS-AES, which can be easily broken using brute-force attacks.

- **Poor Implementation of BitLocker**: BitLocker is poorly implemented, which makes it vulnerable to various attacks, including the TPM bypass attack and the boot process attack.
- **Lack of Security Awareness**: many users are not aware of the security risks associated with BitLocker and do not take adequate steps to protect their data.

It is mentioned that the attack is possible because of the availability of low-cost hardware devices that can be used to bypass BitLocker's security features.

In terms of hardware this attack is also possible because the LPC bus related to TPM communication is not encrypted. This means that an attacker who has physical access to the computer can easily monitor the data that is being sent over the bus.

D. lpc bus

The LPC (Low Pin Count) bus is a computer bus used on IBM-compatible personal computers to connect low-bandwidth devices to the motherboard, such as the boot ROM, "legacy" I/O devices (integrated into a super I/O chip), and Trusted Platform Module (TPM).

1) Purpose of the LPC Bus in a TPM

The LPC bus is a low-speed, multiplexed, point-to-point bus that is used to connect low-bandwidth devices to the motherboard. The LPC bus is a legacy bus and is no longer used in new computer systems.

The TPM chip is a hardware security module that is used to store cryptographic keys and perform cryptographic operations. The LPC bus is used to send commands to the TPM chip and to receive responses from the TPM chip. Some key details:

- The LPC bus is a low-speed bus that operates at a speed of 33 MHz.
- The LPC bus is a multiplexed bus, which means that it uses the same wires to send data in both directions.
- The LPC bus is a point-to-point bus, which means that it connects only two devices.
- The LPC bus is a legacy bus, which means that it is no longer used in new computer systems.

2) Some Other Uses of the LPC Bus in Computer Systems

- Connecting low-bandwidth devices to the motherboard, such as the boot ROM and the BIOS ROM
- Connecting legacy ISA devices to the motherboard
- Connecting Trusted Platform Modules (TPMs) to the motherboard
- Connecting other low-bandwidth devices to the motherboard, such as serial ports and parallel ports

3) BitLocker Extraction

To extract the BitLocker key from a TPM using the LPC bus, an attacker would need to:

- **Gain physical access to the computer**. This could be done by stealing the computer or by gaining access to it through social engineering or other means.
- **Open the computer case and locate the TPM chip**. The TPM chip is usually located on the motherboard.
- **Connect a logic analyzer or other hardware device to the LPC bus**. This will allow the attacker to monitor the data that is being sent over the bus.
- **Boot the computer and wait for the BitLocker key to be sent over the LPC bus**. The BitLocker key is sent from the TPM chip to the operating system when the computer is booted.
- **Capture the BitLocker key using the logic analyzer or other hardware device**. Once the BitLocker key has been captured, the attacker can use it to decrypt the BitLocker-encrypted drive.

4) LPC Security

The LPC bus does not protect the TPM chip from security attacks. In fact, the LPC bus is a potential attack vector that can be used to extract the BitLocker key from the TPM chip.

An attacker could use a hardware device to connect to the LPC bus and monitor the data that is being sent between the TPM chip and the computer's motherboard. This data includes the BitLocker key. Once the attacker has captured the BitLocker key, they can use it to decrypt the BitLocker-encrypted drive.

To protect against this attack, users should enable BitLocker's "TPM-only" mode. This mode requires the TPM chip to be present and functional in order to decrypt the BitLocker-encrypted drive. This makes it much more difficult for an attacker to extract the BitLocker key from the TPM chip.

E. TPM Bypass Attack Demonstration

The TPM Bypass Attack Demonstration is a practical demonstration of how an attacker can bypass the Trusted Platform Module (TPM) chip and extract the encryption key used by BitLocker to encrypt data on a computer. This attack allows the attacker to decrypt the computer's hard drive and access the data without knowing the BitLocker password.

In the video it is used a simple and inexpensive hardware device to perform the attack. The device is connected to the computer's motherboard and allows the attacker to access the TPM chip directly. Once the attacker has access to the TPM chip, they can extract the encryption key and use it to decrypt the computer's hard drive.

It is discussed that several examples of attacks that can be combined to bypass BitLocker

1) TPM Bypass Attack

The TPM bypass attack targets the Trusted Platform Module (TPM) chip, which is a hardware component that is used to store the encryption key for BitLocker. By bypassing the TPM, an attacker can extract the encryption key and decrypt the computer's hard drive.

There are several ways to bypass the TPM, including:

- **Physical Attacks**: An attacker could physically remove the TPM chip from the computer or use a hardware device to access the TPM chip directly.
- **Firmware Attacks**: An attacker could exploit vulnerabilities in the TPM chip's firmware to extract the encryption key.
- **Software Attacks**: An attacker could use a software exploit to bypass the TPM chip and access the encryption key.

2) Boot Process Attack

The boot process attack targets the boot process of the computer. By modifying the boot process, an attacker could prevent BitLocker from loading or could load a malicious version of BitLocker that would allow the attacker to decrypt the computer's hard drive.

There are several ways to modify the boot process, including:

- **Modifying the Bootloader**: An attacker could modify the bootloader to prevent BitLocker from loading or to load a malicious version of BitLocker.
- **Using a Bootkit**: An attacker could use a bootkit to modify the boot process and load a malicious version of BitLocker.
- **Exploiting Vulnerabilities in the Boot Process**: An attacker could exploit vulnerabilities in the boot process to bypass BitLocker.

3) Side-Channel Attacks

Side-channel attacks exploit information that is leaked during the encryption or decryption process. By analyzing this information, an attacker could potentially recover the encryption key. There are several types of side-channel attacks, including:

- **Timing Attacks**: An attacker could measure the time it takes to encrypt or decrypt data and use this information to recover the encryption key.
- **Power Analysis Attacks**: An attacker could measure the power consumption of the computer during the encryption or decryption process and use this information to recover the encryption key.
- **Electromagnetic Attacks**: An attacker could measure the electromagnetic emissions of the computer during the encryption or decryption process and use this information to recover the encryption key.

4) Brute-Force Attacks

A brute-force attack is a type of attack in which an attacker tries all possible combinations of a password or encryption key until the correct one is found. Brute-force attacks can be very time-consuming, but they can be successful if the password or encryption key is weak.

F. Practical Testing and Experimentation'

1) Practical Testing and Experimentation

The author of the video on BitLocker bypass attack conducts practical tests and experiments to evaluate the effectiveness of BitLocker's security features and to demonstrate the TPM bypass attack. These tests and experiments involve setting up test environments, simulating attacks, and analyzing the results to identify potential weaknesses.

2) Test Environments

The author sets up several test environments to simulate different scenarios and configurations. This allows to test the effectiveness of BitLocker's security features in different situations, such as when a computer is booted from a USB drive or when the TPM chip is disabled.

3) Simulated Attacks

The author simulates various attacks on BitLocker, including brute-force attacks, side-channel attacks, and hardware attacks. These attacks are designed to test the strength of BitLocker's encryption algorithms and key management mechanisms.

4) Analysis of Results

This analysis includes examining the time it takes to break BitLocker's encryption, the resources required to carry out the attack, and the impact of the attack on the integrity of the data.

5) TPM Bypass Attack Demonstration

This demonstration shows how an attacker can use a simple and inexpensive hardware device to extract the encryption key from a computer's TPM chip. This demonstration is used to highlight the vulnerability of BitLocker to this type of attack.

The practical testing and provides strong evidence to support the argument that BitLocker can be bypassed using a relatively simple and inexpensive attack.

G. Hardware and software components

1) Hardware Components:

a) TPM Bypass Attack:

- Raspberry Pi 3 Model B+
- Bus Pirate v3.6
- Dupont wires
- Soldering iron
- Solder

b) Boot Process Attack:

- USB flash drive
- Rufus software
- A bootable Linux distribution

2) Software Components:

a) TPM Bypass Attack:

- TPM2-Tools
- Python
- Scapy

3) Boot Process Attack:

- GRUB Customizer
- Syslinux

4) Detailed Explanation per the Attack:

a) TPM Bypass Attack:

- **Hardware Setup**: Connect the Raspberry Pi to the computer's TPM header using the Dupont wires.
- **Software Setup**: Install TPM2-Tools, Python, and Scapy on the Raspberry Pi.
- **Extract the Encryption Key**: Use TPM2-Tools to extract the encryption key from the TPM chip.

b) Boot Process Attack:

- **Create a Bootable USB Drive**: Use Rufus to create a bootable USB drive with a Linux distribution.
- **Modify the Bootloader**: Use GRUB Customizer to modify the bootloader on the USB drive to load a malicious version of BitLocker.
- **Boot from the USB Drive**: Boot the computer from the USB drive.
- **Decrypt the Hard Drive**: The malicious version of BitLocker will decrypt the computer's hard drive.

5) Steps to extract the bitlocker key

- Connect the Raspberry Pi to the computer's TPM header. Use the Dupont wires to connect the Raspberry Pi's GPIO pins to the computer's TPM header.
- Install TPM2-Tools, Python, and Scapy on the Raspberry Pi. Follow the instructions provided by the author in the video.
- Boot the Raspberry Pi.
- Run the following command to extract the encryption key from the TPM chip: **python tpm2_extractkey.py -d /dev/tpm0 -o key.bin**
- The encryption key will be saved to the file key.bin.

H. TPM sniffing

1) TPM Sniffing: Bootmgr Communicates with TPM in the Clear

TPM sniffing is a technique that allows an attacker to extract the BitLocker key from a TPM chip by monitoring the communication between the boot manager and the TPM chip. This is possible because the boot manager communicates with the TPM chip in the clear, meaning that the communication is not encrypted.

2) Purpose of TPM Sniffing

The purpose of TPM sniffing is to extract the BitLocker key from a TPM chip. This key can then be used to decrypt the BitLocker-encrypted drive.

3) How TPM Sniffing Works

TPM sniffing works by monitoring the communication between the boot manager and the TPM chip. This communication takes place over the LPC bus. An attacker can use a hardware device to connect to the LPC bus and monitor the data that is being sent between the boot manager and the TPM chip.

The boot manager is a small program that is responsible for loading the operating system. When the computer is turned on, the boot manager is loaded into memory and it begins to execute. The boot manager then loads the operating system into memory and transfers control to the operating system.

During the boot process, the boot manager communicates with the TPM chip. This communication is used to verify the integrity of the boot process and to load the encryption key for the BitLocker-encrypted drive.

An attacker can use a hardware device to connect to the LPC bus and monitor the communication between the boot manager and the TPM chip. This allows the attacker to extract the encryption key for the BitLocker-encrypted drive.

4) denandz/lpc_sniffer_tpm

The LPC Sniffer TPM is an open-source project that was used to extract BitLocker VMK keys by sniffing the LPC bus when BitLocker was enabled in its default configuration.

The LPC Sniffer TPM is a hardware device that can be used to extract the BitLocker key from a TPM chip by sniffing the communication between the boot manager and the TPM chip. The device connects to the LPC bus and monitors the data that is being sent between the boot manager and the TPM chip.

a) Features of the LPC Sniffer TPM

- I/O read and writes
- Memory read and writes
- Sync errors

b) How to Use the LPC Sniffer TPM

- Modify the EEPROM of the FTDI and enable OPTO mode on Channel B.
- Program lpc_sniffer.bin into your ice40 by iceprog lpc_sniffer.bin.
- *Connect the LPC bus.*
- Extract LPC data: python3 ./parse/read_serial.py /dev/ttyUSB1| tee outlog.
- Extract key from data: cut -f 2 -d' outlog | grep '2...00$' | perl -pe 's/.{8}(..)..\n/$1/' | grep -Po "2c0000000100000003200000(..){32}".

c) Additional Information

- The LPC Sniffer TPM is an open-source project.
- The project was used to extract BitLocker VMK keys by sniffing the LPC bus when BitLocker was enabled in its default configuration.

I. Consequences of the attack

The consequences of the attack discussed in the video are severe and far-reaching:

- **Data Loss**: The attack allows attackers to decrypt and