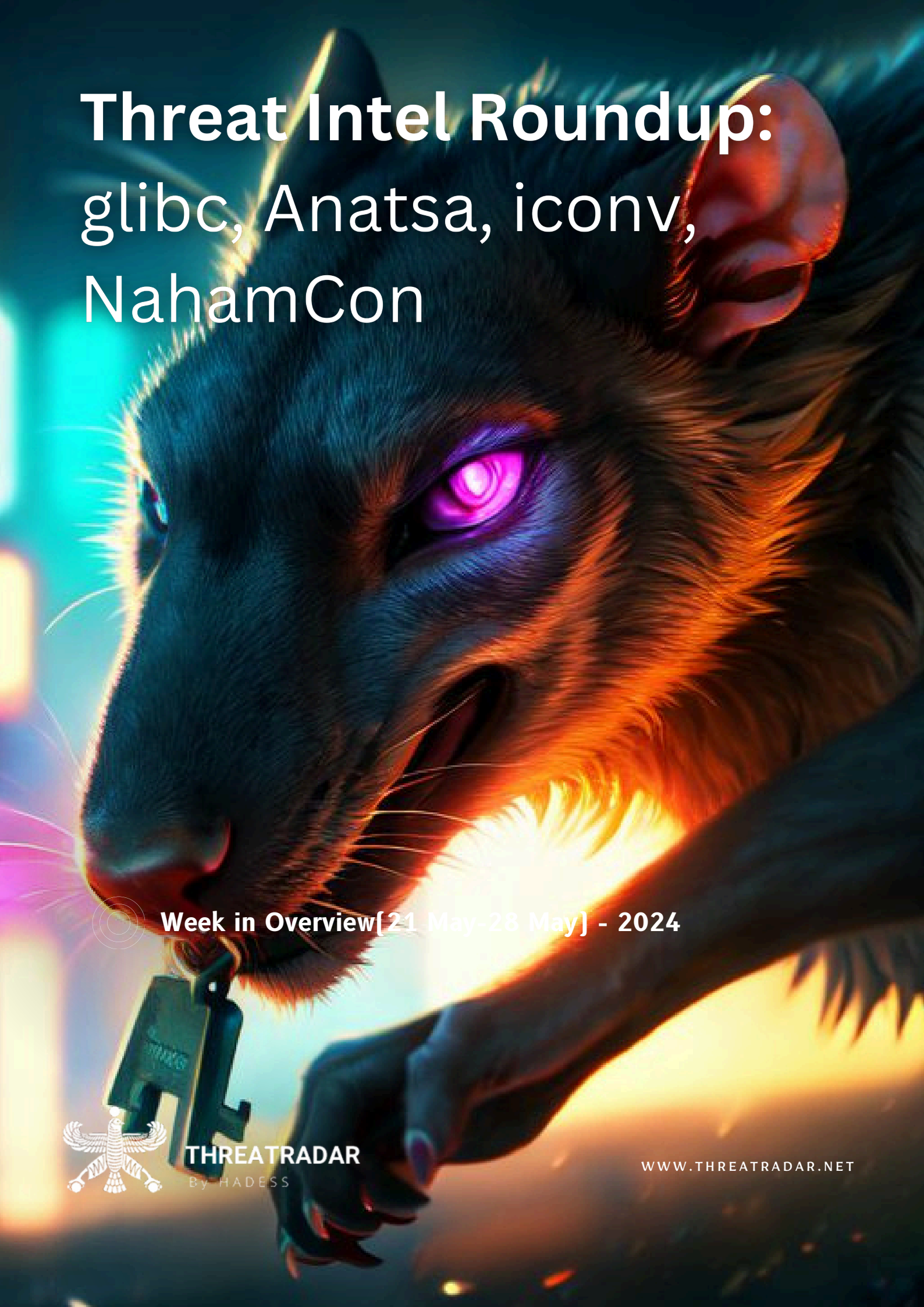
Threat Intel Roundup: glibc, Anatsa, iconv, NahamCon



Week in Overview[21 May-28 May] - 2024

THREATRADAR
By HADESS

WWW.THREATRADAR.NET

Technical Summary

Zero-Interaction Local Privilege Escalation in Zscaler Client Connector

A critical security issue was discovered in the Zscaler Client Connector, enabling a zero-interaction local privilege escalation. By chaining together several low-level vulnerabilities, attackers could escalate privileges and execute arbitrary commands as the NT AUTHORITY\SYSTEM service account on Windows. The vulnerabilities involved:

- CVE-2023-41969: Arbitrary file deletion via ZSATrayManager.
- CVE-2023-41972: Incorrect type validation in the password check for the PERFORM_APP_REVERT function.
- CVE-2023-41973: Lack of input sanitization allowing path traversal in the previousInstallerName parameter.

Exploitation was achieved through bypassing RPC validation checks, password check bypass, path traversal for arbitrary file execution, and DLL hijacking.

Discovery of a New Cluster of Malicious PyPI Packages

A new cluster of malicious packages was identified in the Python Package Index (PyPI) repository. These packages were designed to exfiltrate sensitive information, install backdoors, and perform other malicious activities upon installation. The discovery highlights the growing threat of supply chain attacks targeting open-source repositories. Key findings include:

- Packages with names similar to popular libraries to trick developers.
- Embedded malicious code that executes during the installation process.
- Use of obfuscation techniques to evade detection.

Recommendations include verifying package integrity, using trusted sources, and employing static and dynamic analysis tools to detect malicious behavior.

Exploiting an Ancient Iconv Buffer Overflow Vulnerability in PHP

An ancient buffer overflow vulnerability in PHP's iconv extension was rediscovered, identified as affecting versions that use certain locales. The vulnerability could allow remote code execution if exploited under specific conditions. Technical details include:

- The buffer overflow occurs in the iconv string conversion function when handling malformed input.
- Exploitation involves crafting a special input string that causes a buffer overflow, potentially leading to arbitrary code execution.
- The vulnerability is mitigated in newer PHP versions, but older deployments remain at risk.

Proof-of-concept exploits demonstrate the feasibility of the attack and underscore the importance of updating and patching legacy systems.

Analysis of Anatsa (TeaBot) Malware Campaigns in Google Play Store by Zscaler ThreatLabz

Zscaler ThreatLabz conducted an in-depth analysis of Anatsa (TeaBot) malware campaigns in the Google Play Store. Anatsa is a banking trojan targeting Android devices with sophisticated capabilities. Key observations include:

- The malware disguises itself as legitimate applications to evade detection.
- It employs overlay attacks to steal banking credentials and other sensitive information.
- Advanced evasion techniques, such as dynamic code loading and use of encrypted command-and-control communications.

Key Findings

it is crucial for organizations and individuals to prioritize remediation and patching efforts to safeguard their systems and data. The following key findings highlight the importance of proactive measures to mitigate risks associated with various vulnerabilities and threats:

- Zero-Interaction Local Privilege Escalation in Zscaler Client Connector
- Discovery of a New Cluster of Malicious PyPI Packages
- Exploiting an Ancient Iconv Buffer Overflow Vulnerability in PHP
- Analysis of Anatsa (TeaBot) Malware Campaigns in Google Play Store by Zscaler ThreatLabz
- Technical Details and Proof-of-Concept Exploit for CVE-2024-2961 in GNU C Library

🚨 Vulnerability of the Week

glibc CVE-2024-2961

A critical security flaw identified as CVE-2024-2961 has been discovered in the GNU C Library (glibc), specifically within its iconv() function. This vulnerability can be exploited by threat actors to achieve remote code execution (RCE) on systems that are affected. With a CVSS score of 8.8, this flaw is considered highly severe, highlighting the potential risk it poses to system integrity and security.

The vulnerability resides in the iconv() function of glibc's iconv library. This function is responsible for converting text between different character encodings. The issue particularly affects conversions involving the ISO-2022-CN-EXT character set. When converting from the UCS4 charset, the iconv() function needs to add specific escape characters to indicate changes in the charset. However, the process does not properly check the boundaries of internal buffers, resulting in a buffer overflow that can write up to three bytes beyond the intended memory location.

Security researcher Charles Fol from Ambionics has provided a detailed analysis of this buffer overflow vulnerability. The overflow can be triggered in PHP applications through two primary methods:

1. Direct calls to the iconv() function.
2. Using PHP filters, such as those involved in "file read" vulnerabilities.

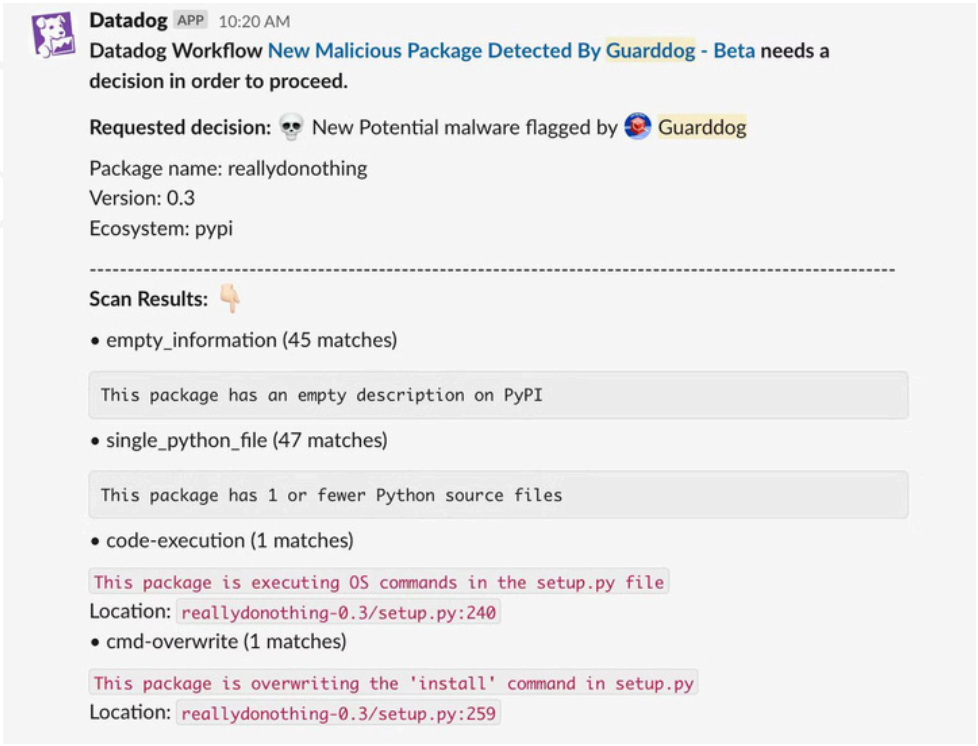
Due to the structure of PHP's heap, this overflow can be exploited to modify part of a free list pointer, ultimately enabling an arbitrary write primitive within the program's memory. Consequently, an attacker with a file read vulnerability and a controlled prefix on a PHP application can achieve RCE. Similarly, forcing PHP to call iconv() with controlled parameters grants the attacker the same capability.

Proof-of-Concept (PoC) Exploit

The proof-of-concept exploit for CVE-2024-2961 involves a series of three requests, which demonstrate the steps an attacker might take to leverage this vulnerability for executing arbitrary commands on the target system:

- Heap and libc Address Extraction:
  - The exploit begins by downloading /proc/self/maps to extract the address of PHP's heap and the filename of the libc in use.
- Downloading libc Binary:
  - The next step involves downloading the libc binary to determine the address of the system() function, which is crucial for executing commands.
- Executing Arbitrary Command:
  - Finally, the exploit triggers the overflow through a crafted request, allowing the execution of an arbitrary command by leveraging the modified memory pointers.

Art of Detection



https://x.com/clintgibler/status/1795168107302166589

As part of our ongoing efforts to secure the software supply chain, we continuously scan newly released PyPI and NPM packages for malicious content. This advisory details our recent identification of a particularly interesting cluster of malicious PyPI packages.

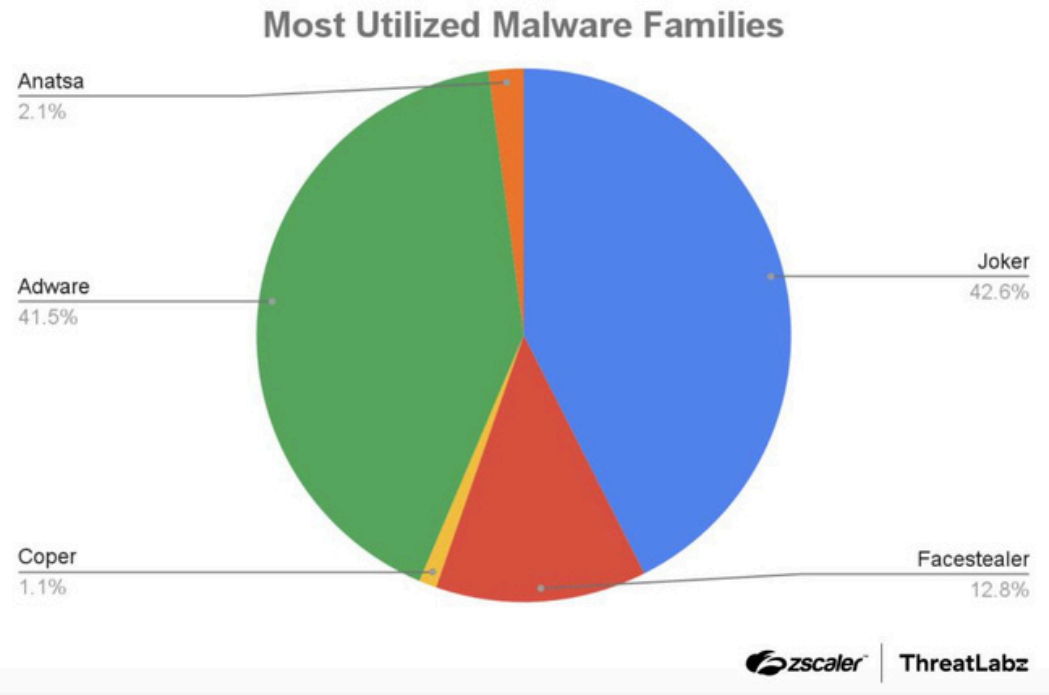
During routine triage, we identified a suspicious PyPI package named reallydonothing, published on May 9, 2024. The package exhibited several red flags:

- Empty Information: The package had an empty description.
- Single Python File: It consisted of a single Python file.
- Command Overwrite: The install command was overwritten to execute code during installation.
- Code Execution: The package executed OS commands.

These indicators triggered our Slack-based triage workflow, prompting further analysis by our researchers.

The package reallydonothing contained a single obfuscated Python file designed to target specific systems. It searches for a secret file whose path, when hashed, matches a predetermined value. If found, the malware downloads and executes a second-stage payload.

😈 Malware or Ransomware



https://x.com/Threatlabz/status/1795108196950102148

This report provides a technical analysis of Anatsa's attack campaigns and an overview of recent trends in the Google Play store.

- Decoy Applications: Threat actors are leveraging seemingly benign applications like PDF readers and QR code readers as droppers to deploy Anatsa malware.
- Disguised Payloads: The second stage payload of Anatsa is disguised as a legitimate application update, deceiving users into installing the malware.
- Evasion Techniques: Anatsa employs various techniques to evade detection, including checks for virtual environments and emulators and corrupting APK ZIP headers to hinder static analysis.
- Global Targets: Anatsa primarily targets banking applications in Europe but has expanded to include institutions in the US, UK, Germany, Spain, Finland, South Korea, and Singapore.

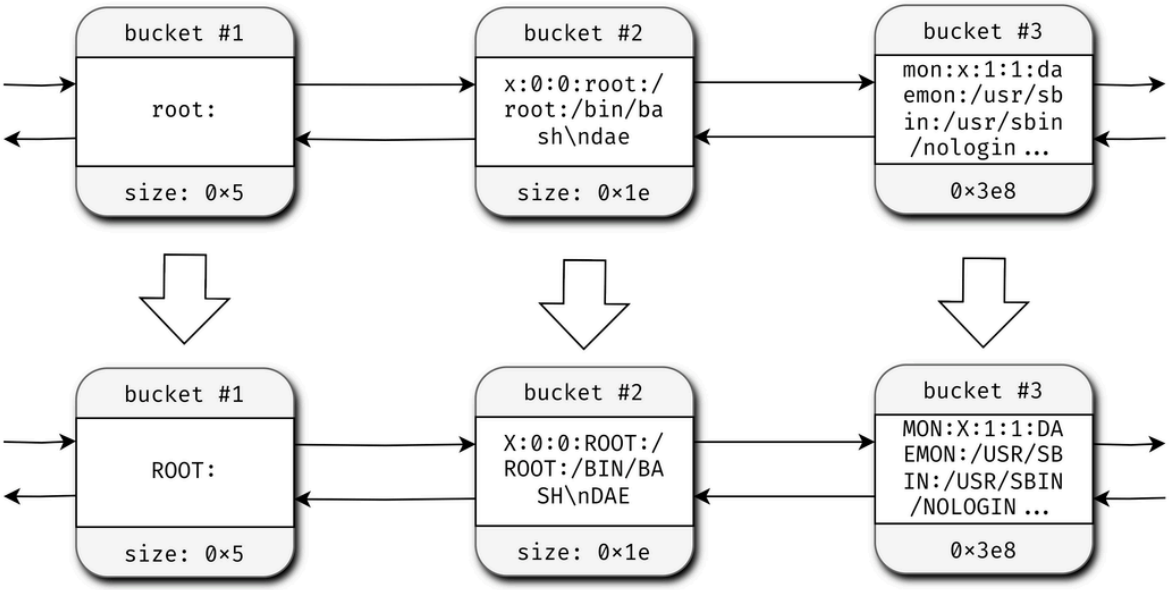
The following sequence illustrates how Anatsa malware is distributed and executed on a victim's device:

1. Initial Infection:
   - The dropper application appears clean and benign when first installed from the Google Play store.
   - Upon installation, the application downloads malicious code or a staged payload from a command-and-control (C2) server, disguised as an application update.
2. Payload Delivery:
   - The dropper application contains encoded links to remote servers, from which the next stage payload is downloaded.
   - The application utilizes reflection to invoke code from the downloaded DEX file.
- Environment Checks:
  - Anatsa performs checks to detect analysis environments and sandboxes.
  - If the device passes these checks, the malware downloads the final payload from the C2 server.
- Data Exfiltration:
  - The malware requests various permissions, including SMS and accessibility options, commonly associated with mobile banking trojans.
  - Anatsa decodes encoded strings and establishes communication with the C2 server.
  - The malware scans the victim's device for targeted banking applications and provides fake login pages to steal credentials.

Technical Analysis

- Malicious Installers: We identified PDF reader and QR code reader applications in the Google Play store acting as installers for Anatsa malware. These applications had amassed over 70,000 installations at the time of analysis.
- Payload and Configuration URLs: The dropper application downloads a payload and configuration file from the remote server to execute the next stage payload.
- Anti-Analysis Techniques: Anatsa uses corrupted ZIP headers in the APK to hinder static analysis. To analyze the payload, these headers need to be fixed alongside the compressed data.
- Permissions and Payload Decryption: Upon execution, the malware decrypts the final DEX payload using a static key embedded within the code.
- Communication with C2 Server: Anatsa establishes communication with the C2 server to register the infected device and retrieve a list of targeted applications for code injection.
- Fake Login Pages: The malware injects fake login pages within a JavaScript Interface (JSI) enabled webview, deceiving users into providing their banking credentials.

1Day



https://x.com/cfreal_

A critical buffer overflow vulnerability has been identified in the iconv() function, existing in glibc since 2000. This vulnerability, CVE-2024-2961, allows attackers to perform a 1-to-3 byte overflow in memory, which can be exploited under specific conditions. Despite the seemingly minimal overflow capability, the vulnerability has far-reaching implications, especially when combined with file read primitives in PHP.

The vulnerability stems from an improper handling of the output buffer in the iconv() function. Specifically, when converting to certain character sets like ISO-2022-CN-EXT, iconv() can write more bytes than specified, leading to a buffer overflow.
The following proof-of-concept demonstrates the overflow:

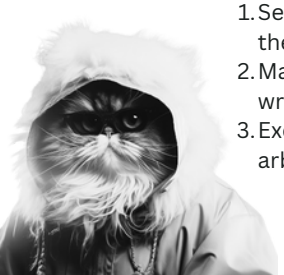
```
$ gcc -o poc ./poc.c && ./poc
Remaining bytes (should be > 0): -1
000000: 41 41 41 41  41 1b 24 2a  48 00 00 00  00 00 00 00    AAAA A.$* H... ....
```

Conditions and Exploitation
For this overflow to be exploitable, two primary conditions must be met:

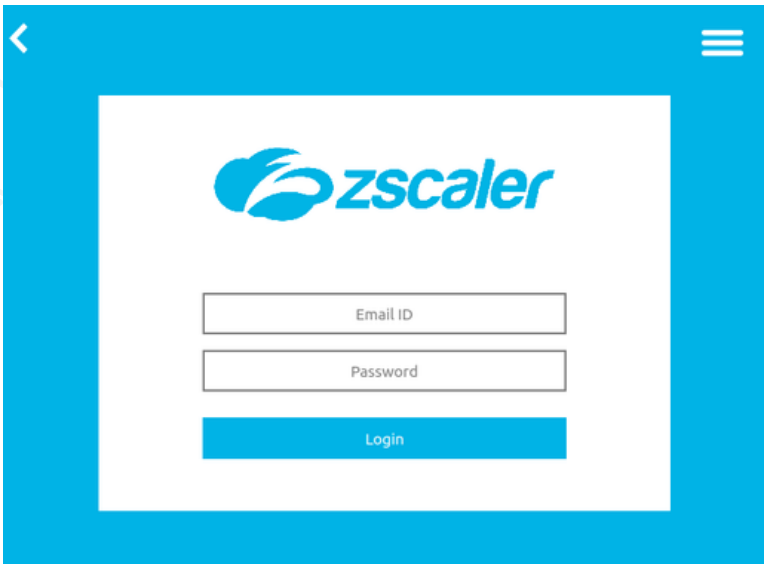
1. The attacker must control the output charset (e.g., ISO-2022-CN-EXT).
2. The attacker must control part of the input buffer to include specific characters.

Initial research into potential targets identified several high-profile libraries and binaries, such as libxml2 and pkexec. While these targets were not directly exploitable due to buffer size management, they highlighted the potential risk.
Exploitation in PHP
The most promising exploitation vector involves PHP's handling of filters and heap memory. By using the convert.iconv filter in PHP, an attacker can manipulate the memory layout and execute arbitrary code. The process involves leveraging PHP's heap management and free list to achieve a controlled memory corruption.
Steps to Exploitation:

1. Setup Heap with Controlled Buckets: Use PHP filters like zlib.inflate and dechunk to create and control the size of memory buckets.
2. Manipulate Free List: Overflow into a free chunk to alter the free list pointers, enabling arbitrary memory write-what-where conditions.
3. Execute Code: Overwrite function pointers in the PHP memory heap to point to system(), enabling arbitrary command execution.



NDay



https://x.com/Dinosn/status/1795137969826742579

Recently, a series of vulnerabilities were discovered in Zscaler Client Connector, leading to a zero-interaction local privilege escalation exploit. This report outlines the discovery, exploitation, and mitigation of these vulnerabilities, which include incorrect type validation, lack of input sanitization, and arbitrary file deletion. The vulnerabilities were chained together to allow a standard user to escalate privileges and execute arbitrary commands as the NT AUTHORITY\SYSTEM service account on Windows.

CVE-2023-41969: ZSATrayManager Arbitrary File Deletion
CVE-2023-41972: Revert Password Check Incorrect Type Validation
CVE-2023-41973: Lack of Input Sanitization Leading to Arbitrary Code Execution

The vulnerabilities were chained together to achieve privilege escalation:

1. RPC Connection Validation Bypass: Exploiting a flaw in the RPC connection validation process via cache grooming and collision.
2. Password Check Bypass: Using incorrect type validation in the PERFORM_APP_REVERT function.
3. Path Traversal: Leveraging the lack of input sanitization to execute arbitrary files.
4. DLL Hijacking: Achieving arbitrary code execution by exploiting the ZSAService binary.

Zscaler Client Connector is a local desktop client used to connect to Zscaler's network tunnels. It consists of two main processes:

- ZSATray: The user-facing frontend application built on .NET Framework.
- ZSATrayManager: A service running as NT AUTHORITY\SYSTEM, handling high-privileged actions.

These processes communicate using Microsoft Remote Procedure Call (RPC).
Vulnerability 1: ZSATrayManager Arbitrary File Deletion (CVE-2023-41969)
Discovered by Winston Ho, this vulnerability allows arbitrary file deletion through ZSATrayManager's RPC interface. Details of this vulnerability can be found in the original Medium blog post.

Vulnerability 2: Revert Password Check Incorrect Type Validation (CVE-2023-41972)
ZSATrayManager does not properly validate the pwdType parameter in the PERFORM_APP_REVERT function, allowing attackers to bypass password checks by specifying an incorrect password type. This flaw can be exploited as follows:

- Incorrect Type Handling: The password check trusts the pwdType passed via the RPC call without verifying if it matches the expected password type.
- Default Password Acceptance: Some password types return true by default if no password has been configured.

The Topic of the Week

https://x.com/NahamSec/status/1790063320206045334

NahamCon 2024 is a virtual security conference that took place from May 23 to May 25, 2024. It's a significant event in the cybersecurity community, featuring a variety of activities including Capture The Flag (CTF) competitions, workshops, and presentations from renowned speakers in the field123.

The conference schedule was packed with insightful talks covering topics such as WordPress hacking, WAF bypass techniques, and practical applications of AI in bounty hunting2. Some notable presentations included:

- "Acing WordPress Hacking with Code Review" by @yeraisci_ & @dhakal_ananda
- "The Art of Bypassing WAFs" by @Brumens2
- "SQL Injection - Tips & Tricks" by @0xTib3rius
- "Modern WAF Bypass Techniques on Large Attack Surfaces" by @infosec_au
- "Writing Caido Plugin Using AI" by @rez0__

The NahamCon CTF 2024 was a 48-hour competition that challenged participants with a range of security-related puzzles and tasks3.

Participants and attendees were encouraged to share their experiences and connect with others using the hashtag #NahamCon2024 on social media platforms12.

Overall, NahamCon 2024 provided a platform for security enthusiasts and professionals to learn, network, and showcase their skills in a collaborative and engaging environment.

cat /etc/HADESS

"Hadess" is a cybersecurity company focused on safeguarding digital assets and creating a secure digital ecosystem. Our mission involves punishing hackers and fortifying clients' defenses through innovation and expert cybersecurity services.

Website:
WWW.HADESS.IO

Threat Radar
WWW.THREATRADAR.NET

Threat Radar is a powerful threat intelligence platform that combines advanced analytics, machine learning, and human expertise to deliver actionable intelligence to organizations. It continuously monitors various data sources, including the deep web, dark web, social media platforms, and open-source intelligence, to identify potential threats, vulnerabilities, and emerging attack patterns.