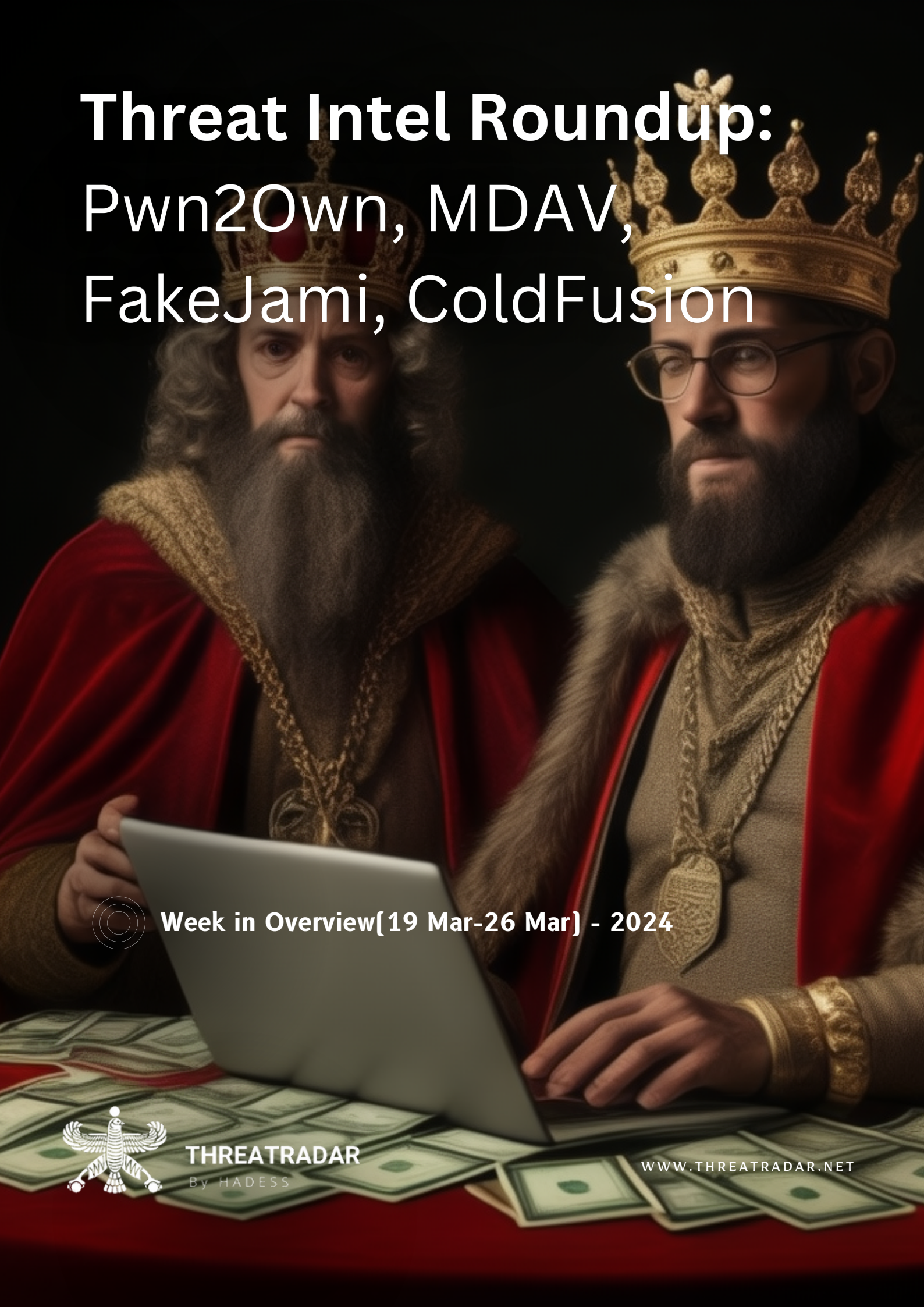
Threat Intel Roundup: Pwn2Own, MDAV, FakeJami, ColdFusion



Week in Overview[19 Mar-26 Mar] - 2024

THREATRADAR
By HADESS

WWW.THREATRADAR.NET

Technical Summary

1.Pwn2Own Vancouver 2024: Pwn2Own Vancouver 2024 showcased the skills of cybersecurity researchers in identifying critical vulnerabilities across various platforms. Notable exploits included escalations of privileges on Windows 11, exploits of VMware Workstation and Oracle VirtualBox, and successful remote code execution on web browsers like Mozilla Firefox, Microsoft Edge, and Google Chrome. A total of $1,132,500 was awarded for 29 unique zero-day exploits, highlighting the importance of prompt vulnerability mitigation by vendors.

2.**SQL Injection in Prepared Statement - CVE-2024–1597**: CVE-2024–1597 exposes a vulnerability in the SQL injection in prepared statements. Attackers can exploit this vulnerability by manipulating numeric and string placeholders in SQL queries. By setting the first parameter to a negative value, attackers can bypass input validation, leading to arbitrary SQL injection in the second parameter. The vulnerability affects versions of software prior to 42.7.2, 42.6.1, 42.5.5, 42.4.4, 42.3.9, 42.2.28, and 42.2.28.jre7, necessitating immediate updates or mitigation strategies.

3.**Leaking ObjRefs to Exploit HTTP .NET Remoting**: This research highlights a method of exploiting HTTP .NET Remoting by leaking object references (ObjRefs). By manipulating exception handling and the LogicalCallContext, attackers can inject malicious code into .NET Remoting servers via HTTP requests. The vulnerability, specific to the Simple query mode, allows attackers to execute arbitrary code remotely, emphasizing the need for thorough security measures in ASP.NET web applications.

4. **New Research: NanoCore (Nancrat) Malware RAT**: NanoCore (Nancrat) is a sophisticated remote access trojan (RAT) managed by various threat actors (TA), advanced persistent threats (APT), and script kiddies. The malware propagates through phishing attacks, leveraging fake links that lead to malicious payloads. Once deployed, NanoCore exhibits persistence, injects itself into processes, collects sensitive information, and communicates with command and control (C&C) servers, posing a significant threat to cybersecurity.

5. **Suspicious MDAV (Microsoft Defender Antivirus) Folder Exclusion Added via Reg.EXE**: This incident involves the suspicious addition of folder exclusions to Microsoft Defender Antivirus (MDAV) via Reg.EXE. The detection query monitors the execution paths for Reg.EXE and flags instances where folder exclusions related to MDAV are added. Such activities may indicate attempts to bypass antivirus protections and require further investigation to mitigate potential security risks.

6. **Security Vulnerability in Adobe ColdFusion (CVE-2024-20767)**: CVE-2024-20767 discloses a security vulnerability in Adobe ColdFusion, allowing attackers to read arbitrary files on affected servers. Exploitation of this vulnerability enables unauthorized access to sensitive information and arbitrary file system writes. Administrators are advised to apply patches promptly and review server configurations to mitigate the risk of exploitation.

7. **FakeJami**: FakeJami is a malware execution chain that utilizes malicious HTA files and PowerShell scripts to download and execute payloads, evading detection by leveraging trusted Windows utilities like mshta.exe and csc.exe. The malware sequence involves contacting malicious domains, downloading payloads, and compiling and executing code, ultimately aiming to steal sensitive information from targeted systems. Detection measures include monitoring execution paths for csc.exe, child processes for HTA files, and creation of .cmdline files, along with blocking known malicious domains and IPs associated with FakeJami.

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:

- Summary of Pwn2Own Vancouver 2024
- SQL Injection in Prepared Statement - CVE-2024–1597
- Leaking ObjRefs to Exploit HTTP .NET Remoting
- New Research: NanoCore (Nancrat) Malware RAT
- suspicious MDAV (Microsoft Defender Antivirus) folder exclusion added via Reg.EXE
- Security Vulnerability in Adobe ColdFusion (CVE-2024-20767)
- FakeJami

🚨 Vulnerability of the Week

ColdFusion CVE-2024-20767

Security researcher ma4ter has uncovered a critical security vulnerability, identified as CVE-2024-20767, within Adobe ColdFusion, a popular web application development platform. This vulnerability poses a significant threat to servers running affected versions of ColdFusion, potentially exposing highly sensitive information to malicious actors. Furthermore, a proof-of-concept (PoC) exploit code has been published, heightening concerns about the exploitation of this flaw.

Overview of the Vulnerability

The vulnerability allows attackers to read arbitrary files on an affected server, enabling them to access highly confidential information without requiring any user interaction. This flaw was initially reported to Adobe, highlighting the importance of prompt mitigation measures to safeguard against potential exploitation.

Attack Methodology

The attack unfolds in two stages:

1. **Retrieving Server UUID**: Attackers first acquire a unique server identifier, known as a UUID, by accessing an API within the ColdFusion Admin panel (/CFIDE/administrator).
2. **Exploiting a Vulnerable Module**: Armed with the obtained UUID, attackers target the Performance Monitoring Toolset (PMS) module, specifically its PMSGenericServlet component. This module facilitates unauthorized access to files stored on the ColdFusion server.

Severity of the Threat

The consequences of a successful exploit are severe:

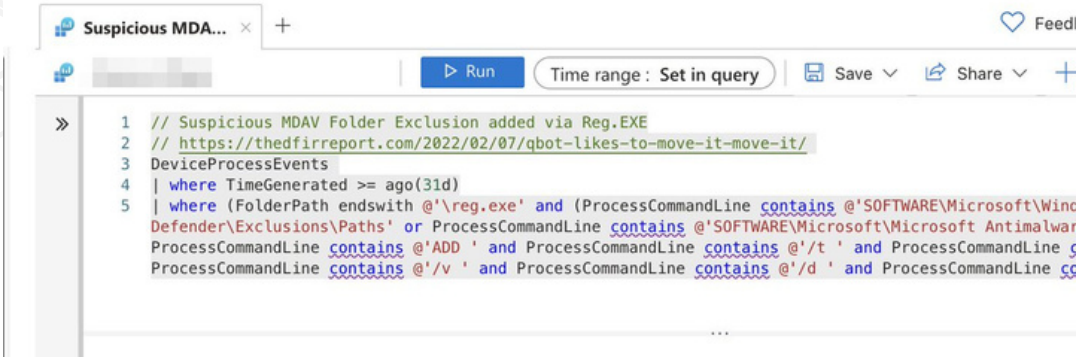
- **Data Theft**: Attackers can exfiltrate a wide range of sensitive information, including confidential data, source code, passwords, and database configurations.
- **System Compromise**: In addition to data theft, attackers can further compromise the system by introducing additional malware or gaining full control over the server. Such actions could potentially jeopardize the entire network connected to the affected server.

Affected Systems

ColdFusion servers running the following versions are vulnerable:

- ColdFusion 2023.6
- ColdFusion 2021.12
- Earlier versions with the Performance Monitoring Toolset enabled and accessible via /pms

Art of Detection



https://twitter.com/ellishlomo/status/1772327445019959542

The detection of a suspicious MDAV (Microsoft Defender Antivirus) folder exclusion added via Reg.EXE indicates potential tampering with antivirus or antimalware configurations on a Windows system. This activity could signify attempts to evade detection by malicious actors or unauthorized modifications made by legitimate users.

Detection Query:
The provided detection query is designed to identify instances where the **reg.exe** utility is used to add folder exclusions to the Windows Defender or Microsoft Antimalware configuration registry keys. Here's a breakdown of the query:

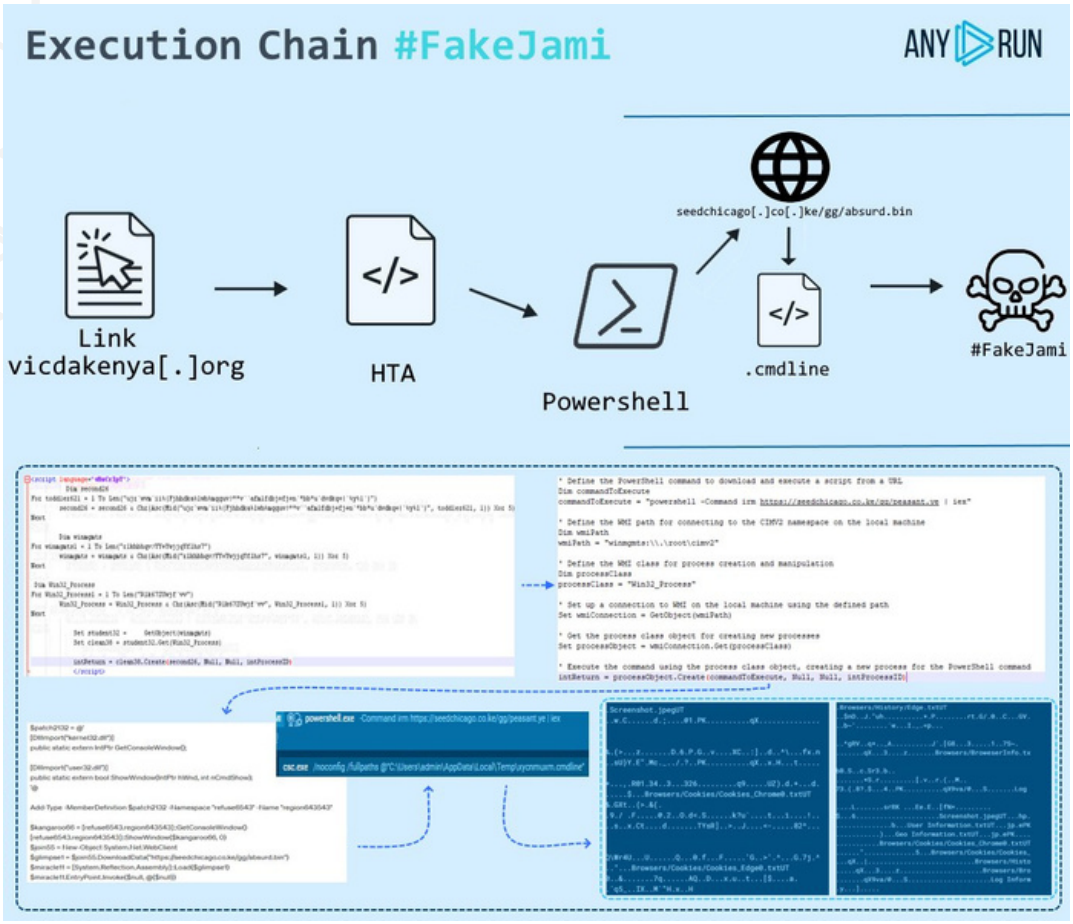
- **DeviceProcessEvents:** This indicates that the query is examining process events on a device.
- **where TimeGenerated >= ago(5d):** Filters events to those that occurred within the last 5 days.
- **where (FolderPath endswith @'\reg.exe':** Specifies that the process event involves the **reg.exe** utility.
- **and (ProcessCommandLine contains @'SOFTWARE\Microsoft\Windows Defender\Exclusions\Paths' or ProcessCommandLine contains @'SOFTWARE\Microsoft\Microsoft Antimalware\Exclusions\Paths'):** Looks for commands modifying the registry keys related to Windows Defender or Microsoft Antimalware folder exclusions.
- **and ProcessCommandLine contains @'ADD ':** Ensures that the command involves adding a registry entry.
- **and ProcessCommandLine contains @'/t ' and ProcessCommandLine contains @'REG_DWORD ':** Checks for specific parameters (**/t** for data type, **REG_DWORD** for registry value type) used in the **reg.exe** command.
- **and ProcessCommandLine contains @'/v ' and ProcessCommandLine contains @'/d ':** Verifies the presence of parameters specifying the value name (**/v**) and data (**/d**) being added.
- **and ProcessCommandLine contains '0'):** Ensures that the value being added is **0**, potentially indicating an exclusion.

Interpretation:
The detection query indicates an attempt to identify suspicious modifications made to Windows Defender or Microsoft Antimalware folder exclusions via the **reg.exe** utility. Such modifications might be indicative of attempts to bypass antivirus or antimalware scans by excluding certain folders from being scanned for malicious content.

Action:
Upon detecting such activity, further investigation is warranted. This may involve:

1. Verifying the legitimacy of the changes made to folder exclusions.
2. Identifying the source and intent behind the modification.
3. Assessing the potential impact on system security and integrity.
4. Taking appropriate remedial actions, such as reverting unauthorized changes and reinforcing security measures.

😈 Malware or Ransomware



https://twitter.com/anyrun_app/status/1772257295680090249

FakeJami is a sophisticated malware strain that employs various techniques to infiltrate and compromise Windows systems, ultimately aiming to steal sensitive information. This malicious software utilizes trusted Windows utilities and obfuscation methods to evade detection and execute its malicious payload.

The execution chain of FakeJami typically begins with the deployment of a malicious HTA (HTML Application) file. HTA files are capable of running scripts on Windows systems and are often used by adversaries to initiate attacks. In the case of FakeJami, this HTA file triggers a PowerShell script to establish communication with a remote server hosted at "seedchicago[.]co[.]ke". From there, it downloads a file named "absurd.bin", which is then fed into "uar3fnt0.cmdline".

The transition to "uar3fnt0.cmdline" is crucial as it prepares the malware for its subsequent actions while evading detection measures. This file serves as an intermediate step in the execution chain, facilitating the seamless progression towards the final payload deployment. To further obfuscate its activities, FakeJami utilizes the C# compiler (csc.exe) to compile "uar3fnt0.cmdline" into an executable format, thereby making it ready for execution.

The ultimate goal of FakeJami is to extract sensitive data from the compromised system. This could include credentials, financial information, or other valuable data. By meticulously orchestrating its execution chain and leveraging legitimate system tools, FakeJami aims to operate covertly while maximizing its chances of success.

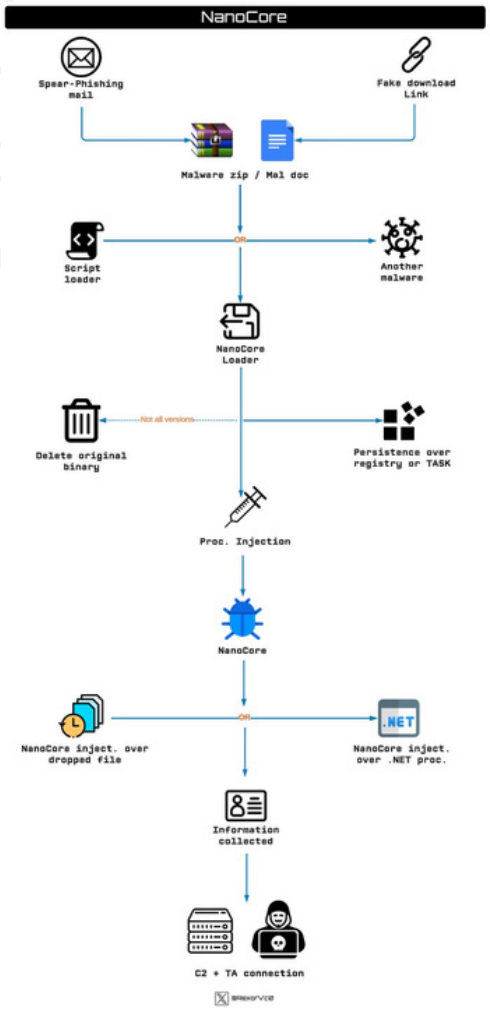
Detecting FakeJami requires a proactive approach that involves monitoring various system activities and behaviors. Key detection options include monitoring the execution paths for csc.exe, tracking child processes associated with HTA files, and keeping an eye on the creation of .cmdline files.

In terms of Indicators of Compromise (IOCs), organizations can watch out for domains like Vicdakenya[.]org and seedchicago[.]co[.]ke, as well as specific IP addresses and file hashes associated with FakeJami. By remaining vigilant and implementing robust detection mechanisms, organizations can mitigate the risk posed by this sophisticated malware.

As exemplified by FakeJami, modern malware threats continue to evolve, employing intricate techniques to bypass security measures and compromise systems. Staying informed about emerging threats and implementing effective security strategies are essential components of defending against such malicious actors.

THREATRADAR
By HADESS

Malware#2



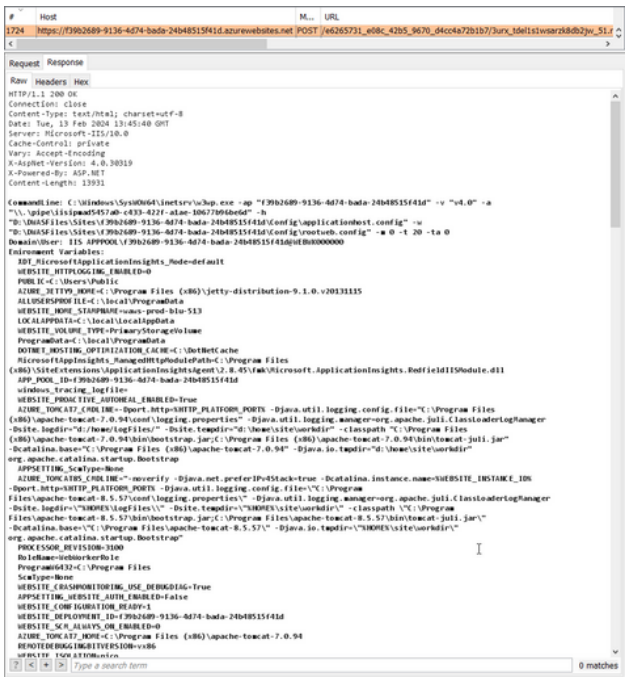
Recent research has shed light on the evolving tactics of the NanoCore (Nancrat) malware Remote Access Trojan (RAT). This sophisticated malware, managed by various Threat Actors (TA), Advanced Persistent Threats (APT), and even script kiddies, continues to pose a significant threat to cybersecurity.

Key Findings:

- **Multiple Handlers:** NanoCore is observed to be managed by several entities, including organized threat actors, advanced hacking groups, and individuals with limited expertise (script kiddies). This diverse range of handlers indicates the widespread use and accessibility of the malware.
- **Phishing Campaigns:** The primary method of NanoCore propagation involves phishing campaigns. Attackers utilize fake links disguised as legitimate sources, leading victims to download ZIP or DOC files containing malicious scripts or loaders. Once executed, these payloads initiate the NanoCore infection process.
- **Payload Execution:** Upon successful execution, NanoCore establishes persistence on the compromised system, ensuring its longevity and resilience against removal attempts. The malware then proceeds with its primary functions, including information collection and communication with Command and Control (C&C) servers.
- **Advanced Capabilities:** NanoCore exhibits advanced capabilities, including data exfiltration, system manipulation, and remote control functionalities. These features enable attackers to extract sensitive information, execute arbitrary commands, and maintain covert access to infected systems.

https://twitter.com/RexorVc0/status/1772524512602415232

1Day



https://twitter.com/binitamshah/status/1772546689267572756

While .NET Remoting has long been considered legacy technology, it continues to persist in various environments due to backward compatibility requirements. Despite the emergence of newer technologies like Windows Communication Foundation (WCF) and .NET Core, .NET Remoting still finds utility in many legacy Microsoft products such as Exchange, SharePoint, and Skype for Business.
In 2022, a blog post titled ".NET Remoting Revisited" explored the internals and risks associated with .NET Remoting. Despite being mostly outdated, the HTTP channel exposed via IIS/ASP.NET remains a default option. This blog post examines this aspect in detail.

HTTP Server Channel Chains

.NET Remoting services via HTTP can be provided through either standalone listeners or integrated within ASP.NET web applications via IIS. The call stack in front of the server channel chain varies depending on the deployment method. In the case of IIS + ASP.NET, the call stack involves components such as HttpRemotingHandlerFactory, HttpRemotingHandler, and HttpHandlerTransportSink.

Leaking ObjRefs

A critical vulnerability arises due to limited exception handling within SoapServerFormatterSink/BinaryServerFormatterSink. If an exception occurs, it is serialized within a ReturnMessage object, potentially containing ObjRef instances. To exploit this vulnerability, two conditions must be met:

1. A way to reach the exception handling in SoapServerFormatterSink/BinaryServerFormatterSink.
2. Some class instance deriving from MarshalByRefObject gets stored in the LogicalCallContext.

Exploitation involves manipulating HTTP headers to pass validation and proceed to the SoapServerFormatterSink/BinaryServerFormatterSink, resulting in exceptions being returned by serialized objects, possibly containing ObjRef instances.

Trust Issues

It's possible to overwrite trusted values returned from HttpRequest with untrusted values from corresponding HTTP headers, potentially bypassing security measures. However, security updates in January 2024 changed the default behavior to prevent such overwriting, enhancing protection against exploitation.

🌶️ Trending Exploit

https://twitter.com/steventseeley/status/1771693634540236952

On March 23, 2024, a critical vulnerability, identified as CVE-2024–1597, was discovered in the PostgreSQL Extended Query Protocol, posing a serious threat to database security. This vulnerability, specific to the Simple query mode, allows for SQL injection attacks when certain conditions are met, potentially leading to unauthorized access, data leakage, and system compromise.

Extended Query Protocol Overview
The PostgreSQL network protocol, known as the Frontend/Backend protocol, includes the extended query protocol, which divides SQL command execution into parse, bind, and execute steps. The mode for executing queries can be specified using the "preferQueryMode" connection property, with "extended" being the default.

Details of CVE-2024–1597
The vulnerability arises under specific conditions:

1. A placeholder for a numeric value immediately preceded by a minus.
2. A second placeholder for a string value after the first placeholder on the same line.
3. Both placeholders must be user-controlled.

When the first parameter is set to a negative value, it effectively comments out the rest of the line, allowing attackers to inject malicious SQL queries into the second parameter. By injecting a newline character ("\n") at the beginning, the second parameter becomes vulnerable to SQL injection attacks.

While prepared statements are a crucial defense mechanism against SQL injection vulnerabilities, developers must adopt a security-first mindset when crafting SQL queries. Properly sanitizing or encoding user input, even within prepared statements, is essential to mitigate the risk of SQL injection attacks. Additionally, updating the PostgreSQL version to one of the patched versions (42.7.2, 42.6.1, etc.) is recommended to address the vulnerability.

The Topic of the Week

https://twitter.com/thezdi/status/1771000382392619297

Pwn2Own Vancouver 2024 concluded with resounding success, showcasing the prowess of cybersecurity researchers and their ability to uncover critical vulnerabilities across various platforms. A total of $1,132,500 was awarded for 29 unique zero-day exploits, making it a highly impactful event in the cybersecurity community.

The title of Master of Pwn was rightfully bestowed upon Manfred Paul, who demonstrated exceptional skill and earned $202,500 along with 25 points. The event also marked the culmination of a successful series, with a combined total of $3,494,750 awarded across the Toronto, Automotive, and Vancouver events in the Pwn2Own calendar for the year.

Top highlights of the event included:

- Marcin Wiązowski's successful escalation of privileges on Windows 11, earning him $15,000 and 3 Master of Pwn points.
- STAR Labs SG's exploit of VMware Workstation, winning $30,000 and 6 Master of Pwn points.
- ColdEye's adept exploitation of Oracle VirtualBox, securing $20,000 and 4 Master of Pwn points.
- Manfred Paul's remarkable sandbox escape of Mozilla Firefox, earning him $100,000 and 10 Master of Pwn points.
- Seunghyun Lee's feat of achieving remote code execution on both Microsoft Edge and Google Chrome, garnering $85,000 and 9 Master of Pwn points.

Throughout the event, participants demonstrated ingenuity and technical prowess, exploiting vulnerabilities across various platforms, including web browsers, operating systems, and virtualization software.

Pwn2Own Vancouver 2024 showcased the critical importance of identifying and addressing vulnerabilities promptly. Vendors now have 90 days to fix the vulnerabilities uncovered during the event, ensuring the continued security of their products.

Special thanks were extended to Tesla for their sponsorship and support, as well as to all the contestants and vendors whose contributions made the event possible.

As the event concluded, anticipation lingered for future editions of Pwn2Own, where cybersecurity researchers will continue to push the boundaries of security and innovation.

HADESS

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.