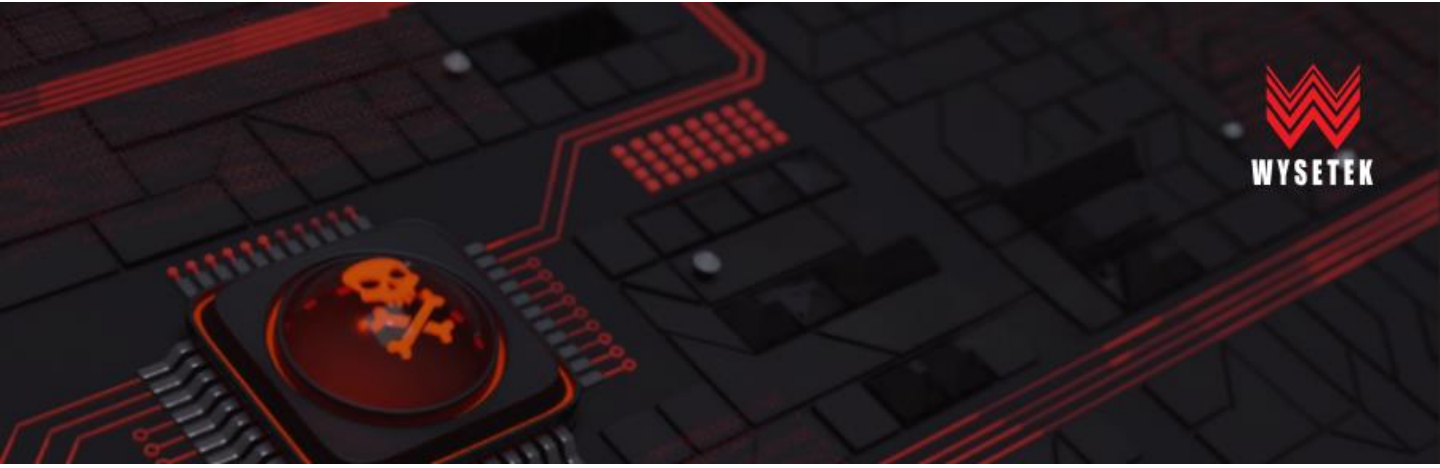


CYBER THREAT ADVISORY

May 2026

TLP: GREEN

Index:

1) Threat Actor: Shiny Hunters

A) EXECUTIVE SUMMARY:

Shiny Hunters is a well-known cybercriminal threat actor group associated with large-scale data breaches, credential theft, extortion campaigns, and unauthorized access to corporate environments. The group gained prominence through attacks targeting organizations across multiple sectors including technology, telecommunications, healthcare, retail, and cloud service providers.

The threat actor is primarily known for:

- Stealing and leaking sensitive customer databases
- Selling compromised data on underground forums
- Credential stuffing and account takeover attacks
- Social engineering and phishing operations
- Exploiting cloud misconfigurations and exposed services

Recent campaigns linked to Shiny Hunters indicate increased activity involving:

- Breach forum advertisements
- Use of stolen credentials for persistence
- Targeting SaaS and cloud-based environments
- Data exfiltration followed by extortion demands

The group often leverages publicly exposed assets, weak authentication mechanisms, and compromised employee credentials to gain initial access. After successful compromise they attempt lateral movement, privilege escalation and extraction of sensitive organizational data. Organizations with exposed internet-facing applications, weak MFA implementation, or poor credential hygiene remain at elevated risk from ShinyHunters-related activities

B) TECHNICAL ANALYSIS

Shiny Hunters is a financially motivated cybercriminal group known for conducting data breaches, credential theft, and extortion campaigns against organizations worldwide. The group primarily targets cloud environments, exposed services, and weak authentication mechanisms to gain unauthorized access to corporate networks.

Common Techniques Used

- Phishing and social engineering attacks
- Credential stuffing using leaked databases
- Exploitation of exposed cloud services and APIs
- Unauthorized access using compromised accounts
- Data collection and large-scale exfiltration

Typical Attack Flow

- Initial access through stolen credentials or phishing
- Persistence using valid accounts and MFA manipulation
- Internal reconnaissance and discovery of sensitive assets
- Collection and exfiltration of customer or corporate data
- Extortion, data leakage, or sale on underground forums

Key Observations

- Heavy reliance on legitimate credentials to evade detection
- Use of HTTPS and cloud services for stealthy exfiltration
- Targeting SaaS and cloud infrastructure environments
- Focus on sensitive customer and business information

Organizations should monitor for unusual authentication activity, unauthorized account changes, suspicious cloud access, and abnormal outbound data transfers to detect potential ShinyHunters-related activity.

S H I N Y H U N T E R S
rooting your systems since '19 ;)

ShinyHunters has breached Instructure (again). Instead of contacting us to resolve it they ignored us and did some “security patches”.

⚠ W A R N I N G

If any of the schools in the affected list are interested in preventing the release of their data, please consult with a cyber advisory firm and contact us privately at TOX to negotiate a settlement. You have till the end of the day by 12 May 2026 before everything is leaked.

Instructure still has until EOD 12 May 2026 to contact us.

C) ECOMMENDATIONS

Security Recommendations – Protection Against Shiny Hunters Activity To reduce the risk associated with Shiny Hunters operations, organizations should implement the following security measures:

Identity & Access Security

- Enforce Multi-Factor Authentication (MFA) across all accounts
- Disable legacy and unused accounts
- Apply strong password policies and password rotation
- Monitor for credential stuffing and brute-force attempts
- Restrict privileged account access using Least Privilege principles

Cloud & Infrastructure Security

- Review cloud configurations and eliminate public exposure
- Secure exposed APIs and administrative portals
- Regularly audit IAM roles, permissions, and OAuth applications
- Enable logging and monitoring for all cloud environments
- Restrict unnecessary external access to critical systems

Monitoring & Threat Detection

- Monitor for unusual login patterns and impossible travel events
- Detect abnormal outbound data transfers and archive creation
- Configure SIEM alerts for suspicious account activity
- Conduct regular threat hunting for credential abuse indicators
- Enable centralized logging for authentication and cloud events

D) INDICATORS OF COMPROMISE

IOC Type	Indicator	Short Detail
IP Address	24.242.93[.]122	Associated with UNC6661 activity (2026)
IP Address	23.234.100[.]107	Linked to suspicious SaaS targeting activity
IP Address	23.234.100[.]235	Observed in ShinyHunters infrastructure
IP Address	73.135.228[.]98	Connected with unauthorized access attempts
IP Address	208.68.36.90	Reported during Sept 2025 campaigns
IP Address	91.199.42.164	Observed in malicious infrastructure
IP Address	23.162.8.66	Linked with suspicious external activity
IP Address	72.5.42.72	Reported in threat intelligence feeds
IP Address	3[.]239[.]45[.]43	Active API usage observed in attacks
Domain	ticket-dior[.]com	Brand impersonation phishing domain
Domain	ticket-lvmh[.]com	Used for social engineering activity
Domain	ticket-louisvuitton[.]com	Malicious impersonation infrastructure
Domain	ticket-nike[.]com	Fake domain targeting users
Telegram Channel	shinygr0up	Threat actor communication channel
Telegram Channel	SLSH6	Associated underground channel
Email Address	shinyc0rp@tuta[.]io	Suspicious actor-linked email
User-Agent	Python/3.11 aiohttp/3.13.1	Used for automated data scraping
Connected App	Gainsight Connected App	Abused for Salesforce API access
Technique	Oauth Abuse	Malicious Oauth apps used for persistence
Technique	Vishing	Phone-based MFA bypass attacks
Technique	API Abuse	Unauthorized Salesforce API access
Log Artifact	ApiAnomalyEvent	Suspicious API activity indicator
Log Artifact	ReportAnomalyEvent	Indicates abnormal report access

2) Notepad++ Vulnerabilities Enable Arbitrary Code Execution

A) Executive Summary

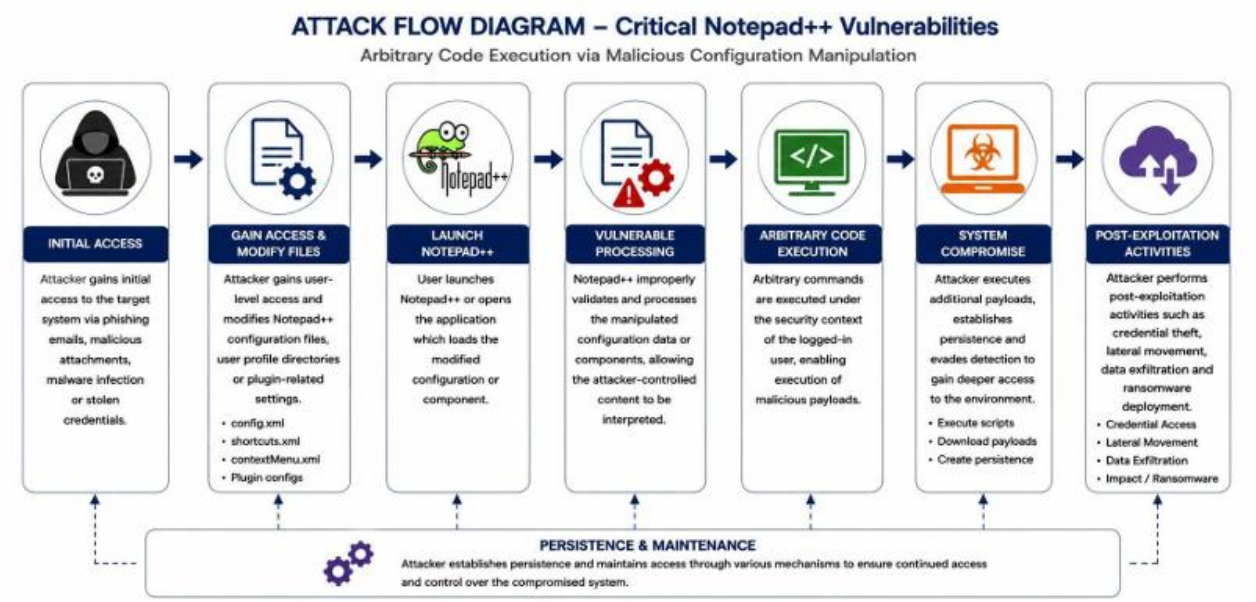
Multiple critical vulnerabilities have been identified in Notepad++, one of the most widely used text and source code editors for Windows. The flaws could allow attackers to execute arbitrary code on affected systems through malicious manipulation of application configuration files and plugin-related components. Because Notepad++ is commonly installed on developer workstations, administrator systems and enterprise endpoints, successful exploitation could provide attackers with a foothold for malware deployment, persistence, credential theft and lateral movement.

The vulnerabilities affect Notepad++ versions up to 8.9.6 and have been addressed in version 8.9.6.1. Organizations are strongly advised to update affected installations immediately and monitor systems for suspicious activity associated with Notepad++ processes.

B) Technical Analysis

The disclosed vulnerabilities stem from improper validation and handling of configuration files and application components used by Notepad++. An attacker with the ability to modify application configuration files user profile directories or plugin-related settings can potentially trigger arbitrary command execution when the application is launched.
The attack does not necessarily require exploitation of a memory corruption vulnerability instead, it abuses trusted application functionality and configuration mechanisms. By manipulating configuration data, an attacker may execute malicious payloads under the security context of the logged-in user.

In enterprise environments, this technique may be used following an initial compromise through phishing, malware infection or stolen credentials. Once access is obtained, threat actors can leverage the vulnerable application to maintain persistence, execute additional payloads, evade detection and establish deeper access within the environment.
Given the widespread use of Notepad++ among IT administrators, developers and privileged users, exploitation may significantly increase the impact of an initial compromise.



C) MITRE ATT&CK Techniques

Initial Access – Phishing (T1566)
Attackers may leverage phishing emails or malicious downloads to gain initial access.

User Execution (T1204)
A user launches the manipulated Notepad++ application or opens a maliciously modified configuration.

Exploitation for Client Execution (T1203)
Exploitation of vulnerable application functionality to execute attacker-controlled code.

Command and Scripting Interpreter (T1059)
Execution of PowerShell, CMD or scripting engines following successful exploitation.

Persistence via Scheduled Tasks (T1053)
Creation of scheduled tasks to maintain long-term access.

Create or Modify System Process (T1543)
Establishment of malicious services or startup mechanisms.

Defense Evasion (T1562)
Disabling security controls or evading endpoint monitoring solutions.

Lateral Movement (T1021)
Use of administrative protocols to move across the environment.

D) Indicators Of Compromise (IOCs)

At present, no publicly available deterministic IOCs such as malicious IP addresses, domains or file hashes have been associated with active exploitation campaigns.

Organizations should monitor for the following suspicious indicators:

1. Unexpected modifications to Notepad++ configuration files such as:
    - config.xml
    - shortcuts.xml
    - contextMenu.xml
    - Plugin configuration files

2. Unusual child processes spawned by notepad++.exe

3. Execution of:
    - powershell.exe
    - cmd.exe
    - wscript.exe
    - cscript.exe
    - mshta.exe
    - rundll32.exe

4. Notepad++ launching immediately before suspicious script or malware execution.

5. Unauthorized changes to user profile directories containing Notepad++ settings.

6. Creation of scheduled tasks, startup entries or persistence mechanisms shortly after Notepad++ execution.

7. Security tool alerts involving suspicious command execution originating from Notepad++.

E) Remediation

Upgrade Immediately
Update all installations to Notepad++ version 8.9.6.1 or later.

Restrict Configuration File Access
Limit write permissions to Notepad++ configuration and plugin directories.

Application Control
Implement Microsoft AppLocker or Windows Defender Application Control (WDAC) policies to restrict unauthorized executable and script execution.

Implement Least Privilege
Ensure users operate with standard privileges and avoid running Notepad++ as Administrator unless required.

F) Conclusion

The newly disclosed Notepad++ vulnerabilities demonstrate how trusted and widely deployed applications can become effective attack vectors when configuration and execution controls are insufficient.

Although exploitation generally requires some level of access to modify application settings or files, successful attacks can enable arbitrary code execution, malware deployment and persistence on affected systems. Given the extensive use of Notepad++ within enterprise environments, exploitation could provide threat actors with an effective mechanism for post-compromise activities.

Organizations should prioritize:

- Immediate patching of affected Notepad++ installations.
- Monitoring for suspicious child processes spawned by Notepad++.
- Restricting unauthorized modifications to application configuration files.
- Enforcing least privilege and application control policies.
- Conducting proactive threat hunting for abnormal Notepad++ activity.

A defense-in-depth strategy combining prompt patch management, endpoint monitoring and application hardening remains essential for mitigating the risk posed by these vulnerabilities.

3)Copy Fail: Universal Linux Local Privilege Escalation Vulnerability

A) Executive Summary

A critical vulnerability identified as CVE-2026-31431 (“Copy Fail”) has been discovered in the Linux kernel, enabling attackers to escalate privileges from a low-level user account to full root access. The vulnerability is particularly severe due to its low exploitation complexity and wide impact, affecting nearly all Linux kernel versions released since 2017.

The flaw allows attackers to manipulate the page cache, which represents the in-memory version of files, rather than modifying files directly on disk. This makes traditional detection mechanisms such as file integrity monitoring ineffective.

By targeting privileged binaries like /usr/bin/su, attackers can inject malicious behavior at runtime and gain unauthorized administrative control.

Additionally, this vulnerability introduces risks in containerized environments, where attackers may exploit shared resources to escape containers and compromise host systems. Given the delayed patch rollout across several major Linux distributions, organizations remain exposed and must take immediate action to mitigate risk.

B) Technical Analysis

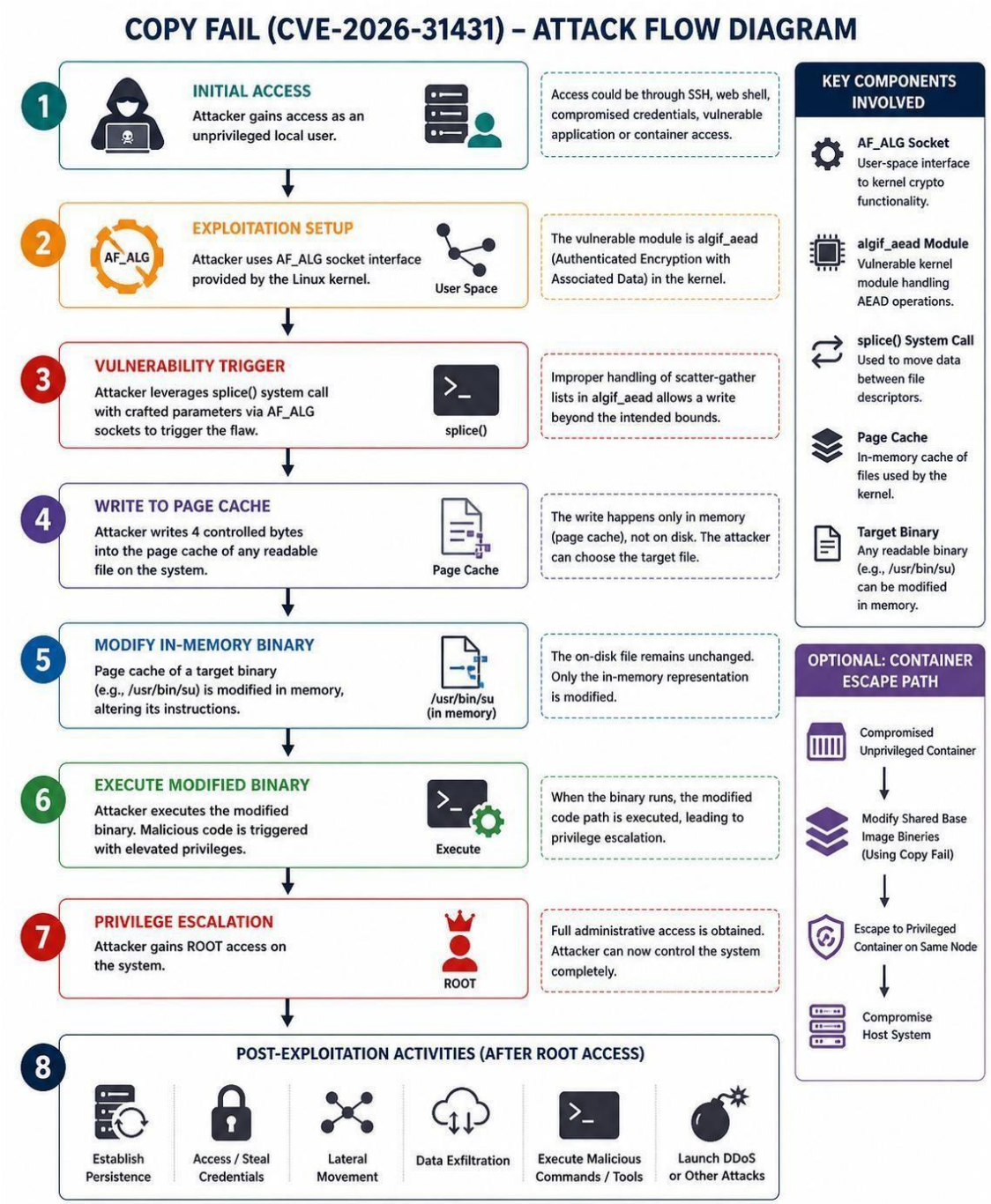
The Copy Fail vulnerability originates from improper handling of memory structures within the Linux kernel's AEAD cryptographic subsystem (algif_aead). Specifically, a flaw in the handling of scatter-gather lists allows controlled memory corruption.

In a typical attack scenario, an unprivileged user leverages AF_ALG sockets in combination with the splice() system call to write controlled data into the page cache of any readable file. Although the write is limited to 4 bytes, it is sufficient to alter the behavior of critical binaries when executed.

Unlike traditional attacks that modify files on disk, this technique alters the in-memory representation of executables. As a result, the malicious modification is temporary and may not leave persistent forensic evidence, making detection significantly more challenging.

Furthermore, in containerized environments, attackers can exploit shared kernel behavior or base images to propagate the attack beyond the initial container boundary. This introduces the possibility of container escape and host compromise, especially in multi-tenant environments.

C) Attack Flow Diagram



D) Affected Products

The vulnerability impacts Linux systems running kernel versions developed between 2017 and the patched release in 2026. Due to the widespread use of Linux across enterprise servers, cloud platforms and embedded systems, the exposure is extensive.

Several distributions including Ubuntu LTS and Debian stable versions have not yet fully released patches at the time of this advisory. While some rolling-release distributions and upstream kernels have incorporated fixes, many enterprise environments relying on stable versions remain vulnerable.

The inconsistent patch availability across vendors increases the risk window, particularly for organizations that depend on delayed update cycles or long-term support releases.

Vulnerable / Not Fully Patched: Ubuntu 18.04 LTS, Ubuntu 20.04 LTS, Ubuntu 22.04 LTS, Debian 12 Bookworm, Debian 11 Bullseye, CloudLinux.

Patching in Progress: Red Hat Enterprise Linux (RHEL 8, RHEL 9), Fedora, SUSE SLES (SLES 12, SLES 15 based on service pack), Amazon Linux 2, Amazon Linux 2023

Patched / Likely Patched: Debian Sid, Debian Forky, Arch), Upstream Linux Kernel.

E) MITRE ATT&CK Techniques

This vulnerability aligns with multiple tactics and techniques defined in the MITRE ATT&CK framework.

Primarily, it falls under Privilege Escalation (T1068), as it allows attackers to gain elevated privileges through exploitation of a system flaw.

It also relates to Execution (T1059), as attackers execute modified binaries to trigger malicious behavior and Defense Evasion, since the attack operates in memory and avoids traditional file-based detection controls.

Additionally, in containerized or shared environments, it may contribute to Lateral Movement, where attackers extend access beyond the initially compromised system.

F) Indicators Of Compromise (IOCs)

Detecting exploitation of Copy Fail can be challenging due to its memory-based nature. However, several indicators can help identify suspicious activity.

One key indicator is the presence of kernel log entries such as “NET: Registered PF_ALG protocol family”, particularly when they occur well after system boot. While this message can appear during normal operations, delayed occurrences may indicate exploitation attempts.

Another observable sign is the execution of suspicious commands, such as downloading exploit code from known sources. Additionally, abnormal behavior in authentication logs may occur. For example, when the su command is executed, logs may show missing or incomplete user information, indicating possible corruption of runtime state.

It is important to note that these indicators are not definitive on their own. Some exploit variants may leave minimal or no traces, making it essential to correlate multiple signals during investigation.

G) Remediation

The most effective mitigation is to update the Linux kernel to a version that includes the official patch.

Organizations should prioritize patching based on risk exposure, focusing on systems that handle untrusted input or operate in multi-user environments.

In cases where patches are not yet available, temporary mitigations can reduce the attack surface.

Disabling the vulnerable module (algif_aead) prevents exploitation through the affected pathway. Additionally, restricting the creation of AF_ALG sockets using seccomp policies can further limit attacker capabilities.

While these measures do not fully eliminate the risk, they provide interim protection until permanent fixes are applied.

H) Conclusion

The Copy Fail vulnerability highlights a critical weakness in Linux systems that can be exploited with minimal effort yet result in complete system compromise. Its ability to bypass traditional detection mechanisms and operate at the memory level makes it particularly dangerous.

As Linux continues to underpin modern infrastructure, including cloud platforms and container ecosystems, vulnerabilities of this nature have far-reaching implications.

Organizations must act swiftly to patch affected systems, implement temporary safeguards, and strengthen monitoring capabilities.

A proactive and layered security approach is essential to defend against evolving threats leveraging vulnerabilities like Copy Fail.

4) RondoDox Campaign: Distributed Botnet Targeting Internet-Facing Services

A) Executive Summary

A malware campaign identified as RondoDox botnet attributed to APT41 is actively targeting exposed internet-facing systems to build a large-scale distributed denial-of-service (DDoS) infrastructure. The botnet primarily focuses on Linux-based systems and IoT devices, exploiting weak credentials, misconfigured services and publicly exposed interfaces to gain initial access. This activity reflects a continued trend of opportunistic mass exploitation aimed at expanding botnet scale rather than targeting specific organizations.

Once deployed RondoDox establishes persistence on compromised systems and connects back to its command-and-control (C2) infrastructure, enabling operators to remotely issue attack instructions. The malware is designed with modular capabilities, allowing it to execute multiple DdoS attack vectors including TCP floods, UDP floods and HTTP-based request attacks, making it highly flexible in disrupting target availability.

The campaign is largely opportunistic in nature, focusing on maximizing botnet size to increase aggregate attack bandwidth rather than maintaining stealth or persistence at the individual target level. The resulting infrastructure is then leveraged to launch high-volume DdoS attacks that can significantly degrade or disrupt targeted services.

The impact of RondoDox is especially concerning for organizations in IT & ITES and telecommunications sectors, where service availability is critical. Compromise of edge devices and servers can contribute to widespread network congestion, service outage and degraded performance across affected environments. The activity is assessed to be global in scope, with infections observed across multiple regions.

This campaign underscores the continued security risks associated with unsecured internet-facing systems and IoT deployments, particularly where credential hygiene and exposure management are weak. It reinforces the need for stronger authentication controls, continuous monitoring of exposed services and proactive network hardening to reduce the attack surface exploited by such botnets.

Organizations are advised to prioritize defensive measures including credential rotation, service exposure audits, intrusion detection for anomalous traffic patterns and segmentation of IoT and server environments.

B) Technical Analysis

The RondoDox campaign primarily initiates with attackers scanning the internet for exposed Linux systems and IoT devices leveraging weak or default credentials and vulnerable services such as SSH and Telnet to gain initial access. This brute-force and opportunistic exploitation approach allows threat actors to compromise systems without relying on sophisticated vulnerabilities making poorly secured environments highly susceptible.

Once access is established the attackers deploy a lightweight RondoDox malware binary onto the compromised host. The payload is designed for rapid execution and minimal system resource consumption enabling large-scale infections. To maintain persistence the malware modifies system startup mechanisms such as cron jobs or initialization scripts ensuring continued execution even after system reboots.

After execution RondoDox establishes communication with its command-and-control (C2) infrastructure typically over common network ports to blend with legitimate traffic and evade detection. The infected system maintains a persistent connection to receive commands from the operators effectively integrating the device into the botnet network. The malware is equipped with modular DdoS capabilities enabling it to launch multiple attack vectors including TCP SYN floods UDP floods and HTTP GET/POST floods.

This flexibility allows attackers to target a wide range of services from web applications to network infrastructure maximizing the disruption potential of coordinated attacks. RondoDox also incorporates basic obfuscation techniques such as stripped binaries and minimal identifiable strings to evade static detection mechanisms. However, its operational behavior remains consistent with typical botnet architectures making it detectable through behavioral monitoring and anomaly-based detection systems.

Propagation within the campaign is largely automated and opportunistic. The malware continuously scans additional vulnerable systems and attempts credential-based access enabling rapid expansion across internet-facing infrastructure. This scanning activity often generates identifiable network patterns which can serve as indicators for early detection by security operations teams.

In the final stage compromised systems are utilized as part of a distributed attack infrastructure to generate high-volume DdoS traffic. The effectiveness of the campaign lies not in advanced evasion techniques but in scale and volume leveraging a large number of infected devices to overwhelm targeted services and degrade availability.

C) MITRE ATT&CK Techniques

Initial Access – Exploit Public-Facing Application (T1190)
Attackers exploit vulnerabilities in internet-facing web servers to gain unauthorized access to targeted systems.

Persistence – Server Software Component: Web Shell (T1505.003)
Web shells such as Godzilla or AntSword are deployed on compromised servers to maintain persistent remote access.

Execution – Command and Scripting Interpreter (T1059)
Threat actors execute commands on compromised systems using scripting environments and command-line utilities.

Privilege Escalation – Exploitation for Privilege Escalation (T1068)
Tools such as Print Spoofer are used to gain elevated privileges within the compromised environment.

Credential Access – OS Credential Dumping (T1003)
Credential dumping tools such as Mimi Katz are used to extract usernames passwords and authentication hashes from memory.

Discovery – Network Service Scanning (T1046)
Attackers scan internal networks to identify active systems, open ports and accessible services.

Lateral Movement – Remote Services (T1021)
Stolen credentials are used to access other systems across the network and expand control.

Collection – Data from Local System (T1005)
Sensitive files databases and internal documents are collected from compromised systems.

Exfiltration – Exfiltration Over Web Services (T1567)
Collected data is compressed, encoded and exfiltrated through web-based channels to avoid detection.

D) Indicators Of Compromise (IOCs)

Indicators	Indicators Type	Description
124.198.131.83	Ipv4	Malicious IP Address
135.148.68.54	Ipv4	Malicious IP Address
14.103.145.202	Ipv4	Malicious IP Address
14.103.145.211	Ipv4	Malicious IP Address
154.91.254.95	Ipv4	Malicious IP Address
169.255.72.169	Ipv4	Malicious IP Address
192.159.99.95	Ipv4	Malicious IP Address
192.183.232.142	Ipv4	Malicious IP Address
192.253.248.5	Ipv4	Malicious IP Address
193.26.115.178	Ipv4	Malicious IP Address
193.26.115.195	Ipv4	Malicious IP Address
23.228.188.126	Ipv4	Malicious IP Address
37.32.15.8	Ipv4	Malicious IP Address
38.59.219.27	Ipv4	Malicious IP Address
41.231.37.153	Ipv4	Malicious IP Address
45.125.66.100	Ipv4	Malicious IP Address
45.135.194.11	Ipv4	Malicious IP Address
45.135.194.32	Ipv4	Malicious IP Address
45.135.194.34	Ipv4	Malicious IP Address
45.153.34.156	Ipv4	Malicious IP Address
45.156.87.165	Ipv4	Malicious IP Address
45.8.145.203	Ipv4	Malicious IP Address
45.88.186.32	Ipv4	Malicious IP Address
45.88.186.85	Ipv4	Malicious IP Address
45.92.1.50	Ipv4	Malicious IP Address
45.94.31.201	Ipv4	Malicious IP Address
45.94.31.89	Ipv4	Malicious IP Address
70.184.13.47	Ipv4	Malicious IP Address
74.194.191.52	Ipv4	Malicious IP Address
78.153.149.90	Ipv4	Malicious IP Address
83.150.218.93	Ipv4	Malicious IP Address
99.241.94.234	Ipv4	Malicious IP Address

E) Recommendations

1. Patch Internet-Facing Systems
Regularly update web servers and applications Disable unnecessary services

2. Monitor Credential Dumping Detect tools such as Mimikatz
Monitor suspicious LSASS access

3. Web Shell Detection
Monitor abnormal web server commands Use EDR solutions to detect web shell activity

4. Network Monitoring
Detect unusual outbound traffic Monitor Base64 encoded data transfers

5. Privileged Access Controls Implement Multi-Factor Authentication Apply least-privilege access policies

6. Threat Hunting python.exe suspicious PowerShell usage

F) Conclusion

The RondoDox campaign represents an opportunistic yet impactful cyber threat focused on building large-scale distributed denial-of-service (DdoS) infrastructure by compromising exposed Linux systems and IoT devices. The threat actor demonstrates the ability to rapidly expand botnet size by leveraging weak credentials and unsecured services rather than relying on sophisticated exploitation techniques. Its lightweight design and automated propagation enable efficient scaling across poorly secured internet-facing environments.

The campaign's operations are primarily aimed at maximizing attack bandwidth and service disruption capabilities rather than maintaining long-term stealth or persistence within a single target. By aggregating a large number of compromised devices the botnet can launch high-volume DdoS attacks capable of degrading or disrupting critical services across sectors such as IT ITES and telecommunications.

This activity highlights the ongoing risks associated with misconfigured systems and inadequate credential hygiene especially in edge and IoT environments. Security teams should prioritize continuous monitoring of network traffic enforcement of strong authentication mechanisms and restriction of unnecessary exposed services. Proactive measures such as network segmentation anomaly detection and regular security audits are essential to mitigate the risks posed by scalable botnet campaigns like RondoDox.

5)Windows BitLocker 0-Day Vulnerability Enables Access to Encrypted Drives

A) Executive Summary

A critical security issue has been demonstrated in Microsoft BitLocker that allows attackers with physical access to bypass disk encryption and gain access to protected data without requiring the recovery key.

The attack is enabled by an open-source tool named Yellow Key. Yellow Key automates a downgrade attack that abuses trusted but outdated boot components to extract BitLocker encryption keys directly from memory.

By booting the system into an older, still signed version of the Windows Recovery Environment, attackers can recover the BitLocker Volume Master Key (VMK) and unlock encrypted drives in less than five minutes.

This technique affects modern Windows 11, Windows Server 2022 and Windows server 2025 devices and demonstrates that systems protected solely by TPM-based BitLocker without a startup PIN remain vulnerable when an attacker gains physical access.

Organizations should immediately strengthen physical security controls, enable pre-boot authentication and deploy Secure Boot revocation updates to mitigate this risk.

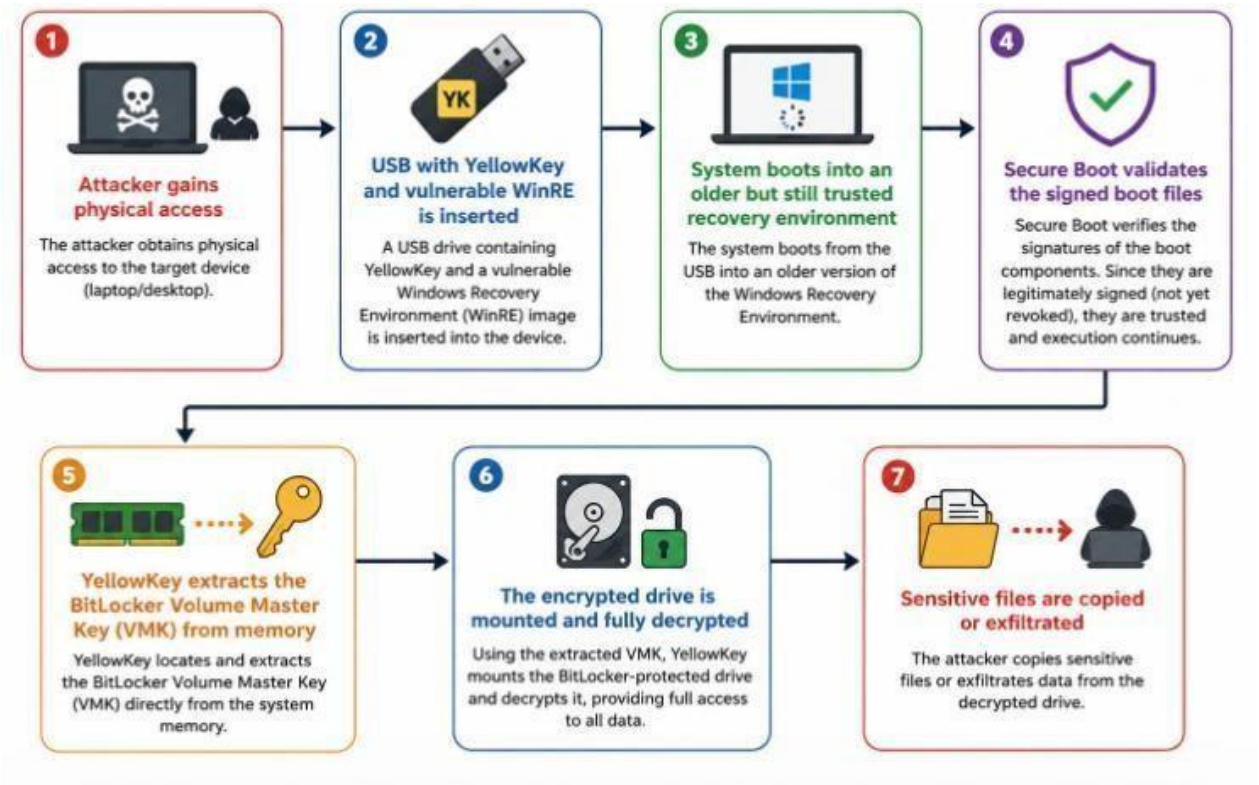
B) Technical Analysis

The attack is performed using Yellow Key an open-source proof-of-concept tool developed by security researcher Nightmare Eclipse.

Yellow Key leverages a weakness in the Windows Boot Manager trust chain. Even after patches are installed, older boot components remain trusted by Secure Boot until Microsoft updates the Secure Boot DBX revocation list.

The tool boots the system using an older signed Windows recovery image, accesses BitLocker-protected volumes, and extracts the Volume Master Key directly from memory. Once obtained, the VMK is used to decrypt and mount the protected drive, providing unrestricted access to all stored data.

The process requires only physical access and a USB drive and leaves minimal forensic traces.



C) MITRE ATT&CK Techniques

Initial Access – Exploit Public-Facing Application (T1190):
Attackers exploit vulnerabilities in internet-facing web servers to gain unauthorized access to targeted systems.

Privilege Escalation (T1068):
Exploitation of a vulnerability to obtain higher-level privileges and bypass BitLocker protections.

Data from Local System (T1005):
Extraction of sensitive files and information from the decrypted local drive.

Unsecured Credentials / Key Access (T1552):
Retrieval of the BitLocker Volume Master Key (VMK) directly from system memory.

Exploitation for Defense Evasion (T1211):
Abuse of trusted but outdated boot components to circumvent encryption and security controls.

Hardware Additions / Removable Media (T1200):
Use of a USB drive containing Yellow Key and a vulnerable Windows Recovery Environment.

D) Indicators Of Compromise (IOCs)

At present, there are no confirmed, deterministic Indicators of Compromise (IOCs) such as file hashes, domains, IP addresses, registry keys or specific event IDs that can reliably identify exploitation of this BitLocker bypass technique.

The attack demonstrated using YellowKey is conducted entirely offline during the pre-boot phase, before the operating system and endpoint security tools are fully loaded. Because the technique extracts the BitLocker Volume Master Key (VMK) directly from memory within the Windows Recovery Environment, it typically leaves little to no persistent forensic evidence on the system.

E) Recommendations

As this is a 0-day vulnerability and no official security patch is currently available to fully eliminate the risk, organizations should focus on compensating controls and hardening measures to reduce the likelihood of successful exploitation.

Enable BitLocker Startup PIN (TPM + PIN)
Configure Microsoft BitLocker to require a pre-boot PIN in addition to TPM protection. This is the most effective mitigation because the encryption key will not be released automatically without user interaction.

Disable Boot from External Media
Prevent systems from booting from USB drives and other removable media through BIOS/UEFI settings, which blocks the primary delivery mechanism used by YellowKey.

Protect BIOS/UEFI with a Strong Administrator Password
Set and securely manage a firmware password to prevent unauthorized users from changing boot order, disabling security features, or re-enabling external boot.

Restrict Physical Access to Devices
Implement strict physical security controls for laptops, desktops, and servers, especially those containing sensitive or regulated information.

Enable Tamper Detection and Chassis Intrusion Alerts
Where supported, activate hardware-based intrusion detection to alert administrators when a device enclosure is opened or modified.

Monitor for Unauthorized Boot and Recovery Activity
Investigate unexpected reboots, use of the Windows Recovery Environment, and changes to BIOS/UEFI settings.

Store Devices Securely When Not in Use
Ensure portable systems are locked in secure storage and are never left unattended in untrusted environments.

Rotate BitLocker Recovery Keys if Compromise Is Suspected
If a device is lost, stolen, or believed to have been accessed physically, regenerate the recovery keys and review the system for evidence of data exposure.

F) Conclusion

The Yellow Key tool demonstrates a practical method for bypassing Microsoft BitLocker by exploiting trusted but outdated boot components. With only physical access and a USB drive, attackers can extract encryption keys and unlock protected drives in minutes.

This vulnerability highlights the limitations of relying solely on TPM-based encryption and reinforces the need for layered security controls, including startup PINs, Secure Boot revocation updates, and strong physical security.

Organizations should assess all BitLocker-protected devices and implement the recommended mitigations immediately to reduce the risk of unauthorized access to sensitive data.