February 13, 2025

CYBERFORCE|Q

FEBRUARY 2025 MONTHLY BULLETIN

CyberForce|Q continues to work diligently to detect the latest threats within the cyber landscape. We are excited to share our most recent Monthly Bulletin to cover the most prominent security incidents over the past month and discuss new Digital Forensics and Incident Response measures used by both adversaries and security professionals.

Key Takeaways

- The malicious node.exe likely originated from a phishing email, containing a malicious attachment that executed the script upon interaction.
- The node.exe process, typically used for running JavaScript applications, was found executing a malicious script from an anomalous directory, indicating potential exploitation or privilege escalation.
- The Evasive Panda malware is delivered through an initial breach, likely via exploitation of vulnerabilities in network appliances or weak SSH credentials.

Case Study

Detection of Malicious node.exe Script with C2 Communication and Persistence Mechanisms

CyberForce|Q's Security Operation Center has observed an increase in malicious payloads being executed via node.exe. In February, a targeted attack was detected involving the execution of a malicious JavaScript payload via node.exe on an endpoint. The executed script demonstrated characteristic behaviors associated with a Remote Access Trojan (RAT) or botnet, including system reconnaissance, persistence mechanisms, and exfiltration readiness. It communicated with command and control (C2) infrastructure over commonly whitelisted ports (80, 443, 1443), and implemented retry logic to maintain connection attempts even after failure. The attack utilized sophisticated evasion techniques to bypass traditional detection methods.

Key Indicators

1. Malicious Execution via node.exe:

- The node.exe process, typically used for running JavaScript applications, was found executing a malicious script from an anomalous directory, indicating potential exploitation or privilege escalation.
- The script was designed to gather system metadata (e.g., OS version, architecture, installed software, IP addresses, etc.), typical of reconnaissance activities seen in RAT infections. This data was likely used to assess the environment for further exploitation or lateral movement.
- The script attempted to establish persistence by modifying startup configurations (e.g., registry keys, scheduled tasks) and triggering re-execution upon system reboot, a common trait of malware seeking long-term control.

2. Command and Control (C2) Communication:

- The script established outbound connections over ports 80 (HTTP), 443 (HTTPS), and 1443, exploiting protocol whitelisting to bypass traditional perimeter defenses. These ports are commonly used for legitimate web traffic, making the traffic harder to detect by firewalls or intrusion prevention systems.
- Predefined C2 domains were identified through network traffic analysis, indicating the script was attempting to establish communication with a known malicious infrastructure. The domains had previously been flagged in threat intelligence feeds, confirming their association with RAT and botnet activity.
- The script exhibited persistent retry behavior, continuously attempting to reconnect to the C2 servers when initial connection attempts failed. This looped retry mechanism is designed to ensure resilience and prevent loss of control over the infected endpoint.

3. Malware Characteristics:

- The script was actively creating registry entries, scheduled tasks, and modifying startup folder locations to ensure re-execution on system reboot. These persistence mechanisms are common tactics used by sophisticated malware to survive system shutdowns and reboots.
- While no immediate data exfiltration was detected, the continuous C2 communication posed a risk for remote control and data exfiltration at a later stage. The persistence and retry behavior suggested the attacker was preparing for future exploitation of the compromised endpoint.

Incident Response:

1. Containment and Isolation:

- Isolate the compromise endpoint(s) from the network immediately to prevent further C2 communication and mitigate the risk of lateral movement.
- Block outbound traffic to the identified C2 domains using perimeter firewall rules to prevent further command-and-control activity.

2. Malware Eradication and System Clean-Up:

- Terminate the malicious JavaScript payload, and remove associated files, including any dropped components from the infected system.
- Identify and delete all persistence mechanisms, such as registry modifications, scheduled tasks, and startup folder entries, to ensure the malware will not be re-executed.
- Conduct a comprehensive system scan to ensure the absence of any remaining malicious artifacts or other compromised files.

3. Determine Root Cause and Exploit Path:

- The attack likely originated from a phishing email, containing a malicious attachment that executed the script upon interaction. The attachment exploited a known vulnerability in a third-party application, allowing the script to execute with elevated privileges.
- Exploit Chain: The attack leveraged a fileless technique by executing the script directly via node.exe, avoiding traditional file-based detection mechanisms.

4. Post-Incident Actions:

- Enhance Email filtering rules to detect and quarantine malicious attachments and phishing emails containing embedded scripts or payloads.
- Review patch management processes to ensure that critical vulnerabilities in third-party applications are addressed, and systems are up to date with the latest security patches.
- Strengthen network segmentation controls to limit the potential for lateral movement and contain any future infections.
- Update threat intelligence feeds and IOC lists to include the C2 domains and associated IP addresses, providing further context for blocking and monitoring future communications.

Recommendations

- Ensure detection mechanisms are configured to monitor for anomalous process execution (e.g., node.exe running from non-standard directories) and unauthorized system modifications (e.g., registry and scheduled task changes). Behavioral-based detection should be prioritized over signature-based detection.
- Use network traffic analysis tools to monitor outbound communications to suspicious domains or ports (especially port 1443). Ensure that traffic to known C2 infrastructure is blocked at the perimeter.
- Strengthen email security measures by implementing attachment sandboxing, phishing detection algorithms, and content filtering to prevent malicious attachments from reaching end-users.
- Implement a rigorous patch management program to address known vulnerabilities in third-party applications that could be exploited to execute malicious scripts or payloads.
- Ensure that incident response protocols are optimized for rapid containment of compromised endpoints and that procedures for C2 traffic blocking, persistence removal, and system restoration are well-practiced.

External Trend

Evasive Panda Malware

Evasive Panda (DaggerFly), a Chinese hacking group, has been leveraging the ELF/Sshdinjector.A!tr malware suite to target network appliances by injecting malicious code into the SSH daemon for persistent access and discreet actions. The malware is delivered through an initial breach, likely via exploitation of vulnerabilities in network appliances or weak SSH credentials. After the device is compromised, a dropper component installs the malware.

Attack Method & Infection Process:

1. Initial Compromise: Exploitation of unpatched vulnerabilities or brute-force attacks targeting weak SSH credentials on network appliances.

2. Dropper Execution: The dropper verifies root access and deploys multiple binaries, including a custom SSH library (libssdh.so), which is injected into the SSH daemon.

3. Persistence & C2 Communication: The injected SSH library establishes command-and-control (C2) communication, enabling the attacker to maintain access and perform malicious actions.

Malware Capabilities:

- System enumeration (hostname, MAC address)
- Credential theft from /etc/shadow
- Process and service listing, log access
- File uploads/downloads, remote shell access
- Remote command execution and persistence maintenance

Recommendations

- **Patch Management:** Ensure SSH services and network appliances are updated to mitigate known vulnerabilities.
- **SSH Hardening:** Disable root login over SSH and enforce strong key-based authentication.
- **Network Segmentation:** Isolate critical infrastructure and limit SSH access to trusted IPs.
- **Monitoring:** Continuously monitor SSH logs for anomalies and unusual binary activity.
- **Endpoint Protection:** Use EDR solutions to detect and block ELF/Sshdinjector.A!tr and similar threats.

How CyberForce|Q Can Help

For 29 years, CyberForce|Q has been a trusted name in advancing cybersecurity programs. Our expertise lies in designing and executing measurable cybersecurity strategies tailored to organizations of all sizes. With a track record of proven results, we offer services such as customized security assessments, robust security operations centers, and comprehensive strategic guidance. Let us assist your organization in prioritizing its goals, elevating your
cybersecurity capabilities, and providing meaningful measurements of progress. Our participants are innovative leaders who share optimal strategies to implement and advance a proven cybersecurity program. CyberForce|Q, in collaboration with our participants, is protecting the cyber realm.

Contact Us For More Information

248-837-1400 • solutions@cyberforceq.com