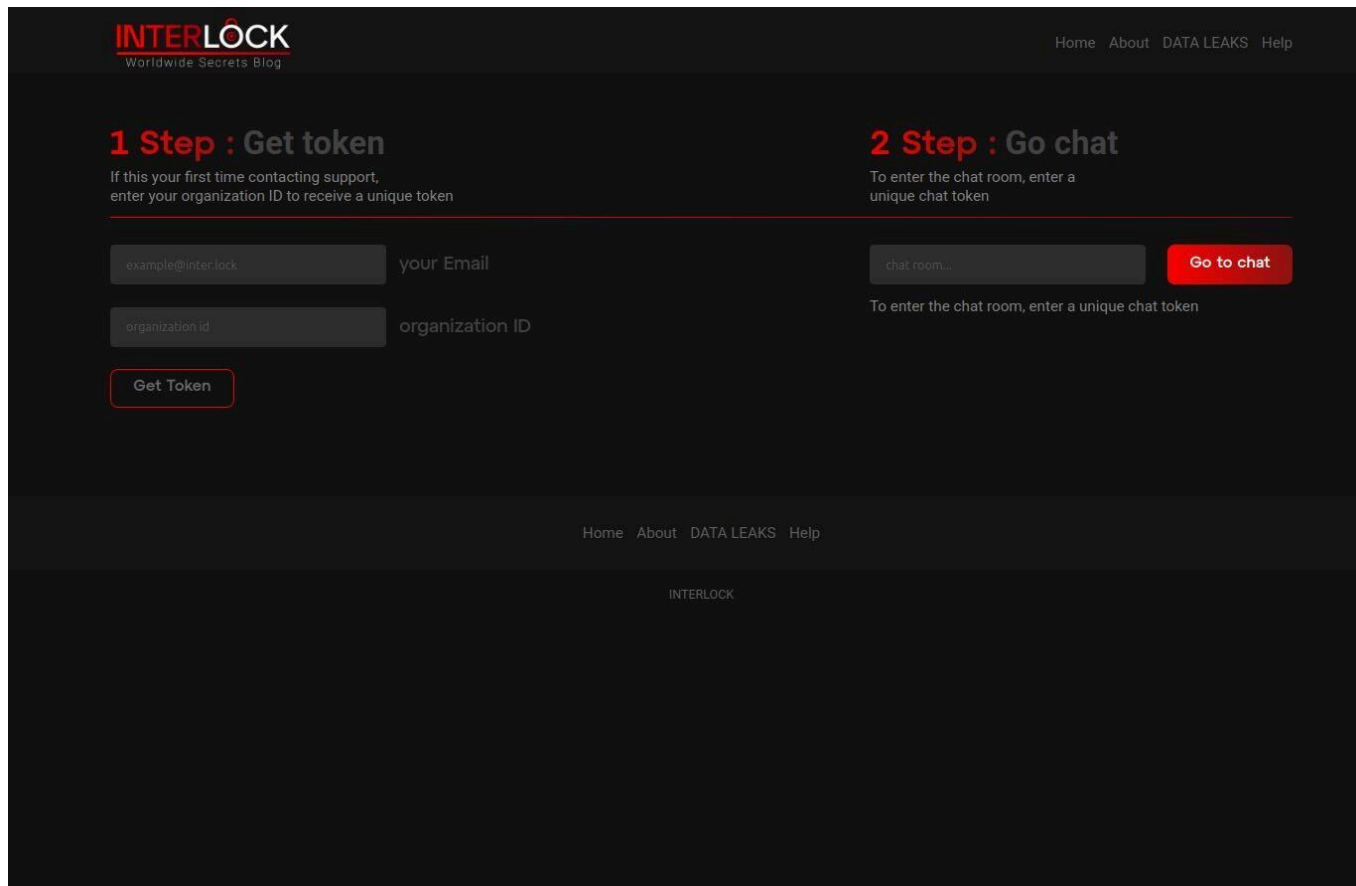


Amazon threat intelligence teams identify Interlock ransomware campaign targeting enterprise firewalls

 aws.amazon.com/ru/blogs/security/amazon-threat-intelligence-teams-identify-interlock-ransomware-campaign-targeting-enterprise-firewalls

March 18, 2026



The screenshot shows the Interlock ransomware website interface. At the top left is the 'INTERLOCK' logo with the tagline 'Worldwide Secrets Blog'. At the top right are navigation links: 'Home', 'About', 'DATA LEAKS', and 'Help'. The main content area is divided into two columns. The left column is titled '1 Step : Get token' and contains the instruction 'If this your first time contacting support, enter your organization ID to receive a unique token'. It features two input fields: 'your Email' (with placeholder 'example@inter.lock') and 'organization ID', followed by a 'Get Token' button. The right column is titled '2 Step : Go chat' and contains the instruction 'To enter the chat room, enter a unique chat token'. It features a 'chat room...' input field and a 'Go to chat' button. Below the input fields in the right column is the text 'To enter the chat room, enter a unique chat token'. At the bottom of the page, the navigation links 'Home', 'About', 'DATA LEAKS', and 'Help' are repeated, followed by the 'INTERLOCK' logo.

Amazon threat intelligence has identified an active Interlock ransomware campaign exploiting CVE-2026-20131, a critical vulnerability in Cisco Secure Firewall Management Center (FMC) Software that could allow an unauthenticated, remote attacker to execute arbitrary Java code as root on an affected device, which was disclosed by Cisco on March 4, 2026.

After Cisco's [disclosure](#), Amazon threat intelligence began research into this vulnerability using Amazon MadPot's global sensor network—a system of honeypot servers that attract and monitor cybercriminal activity. While looking for any current or past exploits of this vulnerability, our research found that Interlock was exploiting this vulnerability 36 days before its public disclosure, beginning January 26, 2026. This wasn't just another vulnerability exploit, Interlock had a zero-day in their hands, giving them a week's head start to compromise organizations before defenders even knew to look. Upon making this discovery, we shared our findings with Cisco to help support their investigation and protect customers.

A misconfigured infrastructure server—essentially, a poorly secured staging area used by the attackers—exposed Interlock's complete operational toolkit. This rare mistake provided Amazon's security teams with visibility into the ransomware group's multi-stage attack chain, custom remote access trojans (backdoor programs that give attackers control of compromised systems), reconnaissance scripts (automated tools for mapping victim networks), and evasion techniques.

AWS infrastructure and customer workloads on AWS were not observed to be involved in this campaign. This advisory shares comprehensive technical analysis and indicators of compromise to help organizations identify potential compromise and defend against Interlock's operations. Organizations running Cisco Secure Firewall Management Center should immediately apply Cisco's security patches and review the indicators provided below.

Discovery and investigation timeline

Amazon threat intelligence identified threat activity potentially related to CVE-2026-20131 beginning January 26, 2026, predating the public disclosure. Observed activity involved HTTP requests to a specific path in the affected software. Request bodies contained Java code execution attempts and two embedded URLs: one used to deliver configuration data supporting the exploit, and another designed to confirm successful exploitation by causing a vulnerable target to perform an HTTP PUT request and upload a generated file. Multiple variations of these URLs were observed across different exploit attempts.

To advance the investigation and obtain additional threat intelligence, we performed the expected HTTP PUT request with the anticipated file content—essentially, we pretended to be a successfully compromised system. This successfully prompted Interlock to proceed to the next stage, issuing commands to fetch and execute a malicious ELF binary (a Linux executable file) from a remote server.

When analysts retrieved the binary, they discovered the same host (attacker-controlled server) is used for distributing Interlock's entire operational toolkit. The exposed infrastructure organized artifacts into separate paths corresponding to individual targets, with the same paths used for both downloading tools to compromised hosts and uploading operational artifacts back to the staging server.

Attribution to Interlock ransomware

The ELF binary and associated artifacts are attributable to the Interlock ransomware family based on convergent technical and operational indicators. The embedded ransom note and TOR negotiation portal are consistent with Interlock's established branding and infrastructure. The ransom note's invocation of multiple data protection regulations reflects Interlock's documented practice of citing regulatory exposure to pressure victims, essentially threatening organizations not just with data encryption, but with regulatory fines and compliance violations. The campaign-specific organization identifier embedded in the note aligns with Interlock's per-victim tracking model.

Interlock has historically targeted specific sectors where operational disruption creates maximum pressure for payment. Education represents the largest share of their activity, followed by engineering, architecture, and construction firms, manufacturing and industrial organizations, healthcare providers, and government and public sector entities.

Temporal analysis performed on timestamps from observed threat activities, artifacts stored on the misconfigured infrastructure server, and metadata embedded within recovered threat artifacts indicates the actor most likely operates in UTC+3 with 75–80% confidence. Systematic analysis across all UTC offsets showed UTC+3 produced the best fit: first activity around 08:30, peak activity between 12:00 and 18:00, and a probable sleep window of 00:30–08:30.

Figure 1: Interlock ransomware negotiation portal where victims enter their organization ID and email address to receive an auth token to begin a negotiation chat session.

Technical analysis: Interlock’s operational toolkit

Post-compromise reconnaissance script

Once Interlock gains initial access, they use a variety of priority tools to complete their attack. Amazon threat intelligence teams recovered a PowerShell script designed for systematic Windows environment enumeration (automated information gathering about the victim’s network). The script collects operating system and hardware details, running services, installed software, storage configuration, Hyper-V virtual machine inventory, user file listings across Desktop, Documents, and Downloads directories, browser artifacts from Chrome, Edge, Firefox, Internet Explorer, and 360 browser (including history, bookmarks, stored credentials, and extensions), active network connections correlated with responsible processes, ARP tables, iSCSI session data, and RDP authentication events from Windows event logs.

The script stages results to a centralized network share (\\JK-DC2\\Temp) using each system’s fully qualified hostname to create dedicated directories—essentially creating a folder for each compromised computer. Following collection, it compresses data into ZIP archives named after each hostname and removes original raw data. This structured per-host output format indicates the script operates across multiple machines within a network—a hallmark of ransomware intrusion chains that prepare for organization-wide encryption.

Custom remote access trojans

Remote access trojans (RATs) are malicious programs that give attackers persistent control over compromised systems, functioning like unauthorized remote desktop software.

JavaScript implant: Amazon threat intelligence recovered an obfuscated JavaScript remote access trojan that suppresses debugging output by overriding browser console methods (hiding its activity from basic detection tools). On execution, it profiles the infected host using PowerShell and Windows Management Instrumentation (WMI), collecting system identity, domain membership, username, OS version, and privilege context before transmitting this data during an encrypted initialization handshake.

Command-and-control communication occurs over persistent WebSocket connections with RC4-encrypted messages using per-message 16-byte random keys embedded in packet headers—essentially, each message uses a different encryption key, making interception more difficult. The implant cycles through multiple operator-controlled hostnames and IP addresses in randomized order with exponential backoff between reconnection attempts.

The implant provides interactive shell access, arbitrary command execution, bidirectional file transfer, and SOCKS5 proxy capability for tunneling TCP traffic (routing malicious traffic through other systems to hide its origin). Self-update and self-delete capabilities allow operators to replace or remove the implant without reinfection, supporting operational cleanup to hinder forensic investigation.

Java implant: A functionally equivalent client implemented in Java provides identical command-and-control capabilities. Built on GlassFish ecosystem libraries, it uses Grizzly for non-blocking I/O transport and Tyrus for WebSocket protocol communication. In simpler terms, Interlock built the same backdoor in two different programming languages, ensuring they maintain access even if defenders detect one version.

Infrastructure laundering script

Sophisticated threat actors don't attack from their own infrastructure, they build disposable relay networks to hide their tracks. Amazon threat intelligence teams identified a Bash script that configures Linux servers as HTTP reverse proxies (intermediary servers that forward traffic to hide the attacker's true location). The script performs system updates, installs fail2ban with SSH brute-force protection, and compiles HAProxy 3.1.2 from source. The HAProxy instance listens on port 80 and forwards all inbound HTTP traffic to a hardcoded target IP, with systemd ensuring persistence across reboots.

A notable component is a log erasure routine running as a cron job every five minutes. The routine truncates all *.log files under /var/log and suppresses shell history by unsetting the HISTFILE variable. This aggressive evidence destruction, wiping logs every five minutes, combined with the purpose-built HTTP forwarding proxy, indicates the script establishes disposable traffic-laundering relay nodes. These nodes obscure exploit traffic origin, relay command-and-control communications, or proxy data exfiltration, making it nearly impossible to trace attacks back to their source.

Memory-resident webshell

Amazon threat intelligence teams observed a Java class file delivered as an alternative to the ELF binary drop. When loaded by the Java Virtual Machine (JVM), its static initializer registers a ServletRequestListener with the server's StandardContext, essentially installing a persistent memory-resident backdoor that intercepts HTTP requests without writing files to disk. This "fileless" approach evades traditional antivirus scanning that looks for malicious files.

The listener inspects incoming requests for specially crafted parameters containing encrypted command payloads. Payloads are decrypted using AES-128 with a key derived from the MD5 hash of the hardcoded seed "geckoformboundary99fec155ea301140cbe26faf55ed2f40" (using the first 16 characters: 09b1a8422e8faed0).

Decrypted payloads are treated as compiled Java bytecode, dynamically loaded into the JVM, and executed—a technique designed to evade file-based detection by running malicious code entirely in memory.

Connectivity verification tool

Amazon threat intelligence teams recovered Java class files implementing a basic TCP server listening on port 45588 (encoded as Unicode character 𐄂 to obscure the port number from static analysis). The server accepts connections, logs connecting IP addresses, sends a greeting message, and immediately closes connections. This operational profile is consistent with a lightweight network beacon—essentially a “phone home” tool used to verify successful code execution or confirm network port reachability following initial exploitation.

Legitimate tool abuse

Interlock deployed ConnectWise ScreenConnect, a legitimate commercial remote desktop tool, alongside custom implants. When ransomware operators deploy legitimate remote access tools alongside their custom malware, they’re buying insurance—if defenders find and remove one backdoor, they still have another way in. This indicates multiple redundant remote access mechanisms—a pattern consistent with ransomware operators seeking to maintain access even if individual footholds are removed. The tool’s legitimate network footprint helps blend with authorized remote administration traffic, making detection more challenging.

Amazon threat intelligence teams also recovered Volatility, an open-source memory forensics framework typically used by incident responders (the same tool defenders use to investigate attacks). While no artifacts indicated automated use, its presence alongside custom implants and reconnaissance scripts is consistent with advanced threat operations. Both ransomware groups and nation-state actors have been observed deploying Volatility during intrusions. The tool’s focus on parsing memory dumps provides access to sensitive data such as credentials stored in RAM, which can enable lateral movement (spreading through the network) and deeper environment compromise in support of ransom operations or espionage objectives.

Interlock also used Certify, an open source offensive security tool designed to exploit misconfigurations in Active Directory Certificate Services (AD CS). For ransomware operators, Certify provides a pathway to identify vulnerable certificate templates and enrollment permissions that allow requesting authentication-capable certificates. These certificates can be used to impersonate users, escalate privileges, or maintain persistent access. These capabilities directly support both initial compromise and long-term persistence objectives in ransomware operations.

Indicators of compromise (IoCs)

The following indicators support defensive measures by organizations that may be affected. Due to Interlock’s use of content variation techniques, most file hashes are not included as reliable indicators. The threat actor modified most artifacts like scripts and binaries downloaded to different targets. This resulted in different file hashes for functionally identical tools. The customization allowed each attack to evade signature-based detection that looks for exact file matches.

206.251.239[.]164	Exploit source IP	Active Jan 2026
199.217.98[.]153	Exploit source IP	Active Mar 2026

89.46.237[.]33	Exploit source IP	Active Mar 2026
Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:136.0) Gecko/20100101 Firefox/136.0	Exploit HTTP User-Agent	Observed Jan 2026 and Mar 2026
b885946e72ad51dca6c70abc2f773506	Exploit TLS JA3	Observed Jan 2026 and Mar 2026
f80d3d09f61892c5846c854dd84ac403	Exploit TLS JA3	Observed Mar 2026
t13i1811h1_85036bcba153_b26ce05bbdd6	Exploit TLS JA4	Observed Jan 2026 and Mar 2026
t13i4311h1_c7886603b240_b26ce05bbdd6	Exploit TLS JA4	Observed Mar 2026
144.172.94[.]59	C2 Fallback IP	Active Mar 2026
199.217.99[.]121	C2 Fallback IP	Active Mar 2026
188.245.41[.]78	C2 Fallback IP	Active Mar 2026
144.172.110[.]106	Backend C2 IP	Active Mar 2026
95.217.22[.]175	Backend C2 IP	Active Mar 2026
37.27.244[.]222	Staging host IP	Active Mar 2026

hxxp://ebhmkoohcccl45qesdbvrjqtyro2hmhkmh6vkyfyjjzflm3ix72aqaid[.]onion/chat.php	Ransom negotiation portal	Active Mar 2026
cherryberry[.]click	Exploit Support Domain	Active Jan 2026
ms-server-default[.]com	Exploit Support Domain	Active Mar 2026
initialize-configs[.]com	Exploit Support Domain	Active Mar 2026
ms-global.first-update-server[.]com	Exploit Support Domain	Active Mar 2026
ms-sql-auth[.]com	Exploit Support Domain	Active Mar 2026
kolonialeru[.]com	Exploit Support Domain	Active Mar 2026
sclair.it[.]com	Exploit Support Domain	Active Mar 2026
browser-updater[.]com	C2 domain	Active Mar 2026
browser-updater[.]live	C2 domain	Active Mar 2026
os-update-server[.]com	C2 domain	Active Mar 2026
os-update-server[.]org	C2 domain	Active Mar 2026

os-update-server[.]live	C2 domain	Active Mar 2026
os-update-server[.]top	C2 domain	Active Mar 2026
d1caa376cb45b6a1eb3a45c5633c5ef75f7466b8601ed72c8022a8b3f6c1f3be	Offensive security tool (Certify)	Observed Mar 2026
6c8efbcef3af80a574cb2aa2224c145bb2e37c2f3d3f091571708288ceb22d5f	Screen locker	Observed Mar 2026

Defensive recommendations

Organizations should take the following actions to protect against Interlock ransomware operations.

Immediate actions:

- Apply Cisco's security patches for Cisco Secure Firewall Management Center
- Review logs for the indicators of compromise listed above
- Conduct security assessments to identify potential compromise
- Review ScreenConnect deployments for unauthorized installations

Detection opportunities:

- Monitor for PowerShell scripts staging data to network shares with hostname-based directory structures
- Detect Java ServletRequestListener registrations in web application contexts (unusual modifications to Java web applications)
- Identify HAProxy installations with aggressive log deletion cron jobs (proxy servers that erase their own logs every five minutes)
- Watch for TCP connections to unusual high-numbered ports (e.g., 45588)

Long-term measures:

- Implement defense-in-depth strategies with multiple layers of security controls
- Maintain continuous threat monitoring and hunting capabilities
- Ensure comprehensive logging with secure, centralized log storage (stored separately from systems that could be compromised)
- Regularly test incident response procedures for ransomware scenarios
- Educate security teams on Interlock's tactics, techniques, and procedures

The real story here isn't just about one vulnerability or one ransomware group—it's about the fundamental challenge zero-day exploits pose to every security model. When attackers exploit vulnerabilities before patches exist, even the most diligent patching programs can't protect you in that critical window. This is precisely why

defense in depth is essential—layered security controls provide protection when any single control fails or hasn't yet been deployed. Rapid patching remains foundational in vulnerability management, but defense in depth helps organizations not to be defenseless during the window between exploit and patch.

Amazon Threat Intelligence teams continue to monitor Interlock ransomware operations and will provide updates as additional information becomes available. The intelligence gathered from this campaign is being integrated into AWS security services to protect customers proactively.

If you have feedback about this post, submit comments in the **Comments** section below. If you have questions about this post, [contact AWS Support](#).

Comments

Log in to comment