ON2IT AND AMS-IX PRESENT: THREAT TALKS

Threat Talks

Breaking the Illusion: Exposing Security Fallacies

The costly mistakes of a false sense of cybersecurity.

threat-talks.com

When you walk into a building with clearly visible security cameras on the outside, numerous bolt locks on the entrance door and maybe even biometric security to get in – you'll likely feel very safe once you're inside.

But what about the back door? Is the same high level of security applied throughout the building, or could someone with ill-intent get in somewhere else?

When it comes to cybersecurity, the cyber equivalents of these high-tech security systems can lead to a false sense of (cyber)security. Will automated updates keep your PC safe from malware and intrusions? Does complying with all relevant frameworks mean your cybersecurity is perfect?

Misconceptions like these are dangerous in the world of cybersecurity; they lead to blind spots that threat actors and hackers are all too happy to take advantage of.

In this **Security Fallacies** episode of Threat Talks, we take a look at common blind spots in cybersecurity and what real-world risks they may conceal.

Are you ready to bust some cybersecurity myths?

In this Threat Talks we discuss the following threats:

- UNIX CUPS
- CrowdStrike
- XZ Utils Library

34% of organizations reported experiencing a cyber-attack that **evaded their antivirus** software in the past year.

Source: ESET

63% of businesses do not have an adequate **cybersecurity budget**.

Source: Source: AT&T's 2022 Cybersecurity Insights Report

avg. cost of adata breach

$ 4.45 million

The average **cost of a data breach** in 2024 reached a new high of **$4.45 million**.

Source: IBM

95% of all cybersecurity incidents were contributed by a **human error** cause.

Source: IBM

UNIX CUPS Remote Code Execution

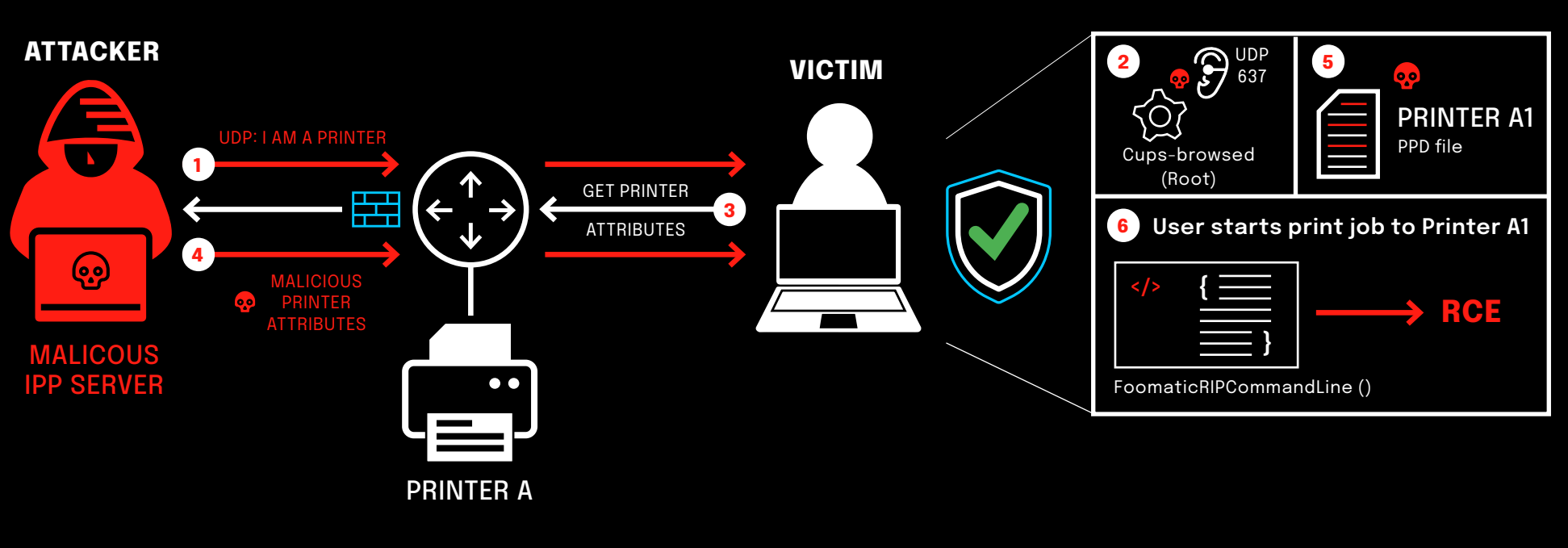
The Fallacy of Default Trust

On September 26, 2024, a set of vulnerabilities affecting the Common UNIX Printing System (CUPS) was disclosed. These vulnerabilities allowed Remote Code Execution (RCE) through the exploitation of multiple flaws in the cups-browsed service and related libraries. Attackers could send crafted UDP packets to port 631, chain the vulnerabilities, and potentially gain full control of a system by executing arbitrary commands.

mSOC confidence score	Confirmed
Threat category	Vulnerability - CVE Disclosures
Severity	High

ATT&CK Technique	Attack Strategy	Evasion	Complexity	Target Type
T1210 - Exploitation of Remote Services	Exploit flaws in packet validation and configuration	Use of legitimate network protocols	High	Any

ATT&CK Mitigation	Attack vector	Detection	Threat level	Threat Actor Type
M1030 - Network Segmentation	Software vulnerability	Network monitoring	High	Any



Malicious printer impersonation

1. The attacker sends a malicous UDP packet to the victim on port 631, impersonating a new printer.

M Access to UDP port 631 should be limitied through firewall rules. Only trusted devices should be able to communicate with the cups-browsed service.

Trust issues

2. The cups-browsed service [1] runs UNIX CUPS Remote code execution by default as root, listens on UDP port 631 and accepts connections from any source. The service does not perform packet validation. This leads to trusting the attacker's spoofed printer announcement.

M Cups-browsed service should be disabled unless strictly necessary. A zero trust approach with network segmentation can significantly reduce the risk of malicious connections. Continuous monitoring of web service calls for abnormal behavior can help detect and prevent such exploits.

Printer Attributes Request

3. The victim, trusting the packet source, will send a Get-Printer-Attributes IPP [2] request to an attacker controlled URL, hosted on a malicious IPP server.

Malicious Payload Injection

4. At this point the attacker responds with a series of malicious attributes, including the payload to be passed to trigger remote code execution.

Printer installation

5. The victim retrieves all the information provided by the attacker and creates a PPD [3] file without sanitizing any input, effectively adding the new printer to the system printers.

Remote code execution

6. Once the victim initiates a print job using the maliciously added printer, the FoomaticRIPCommandLine [4] function processes the PPD file, which contains the attacker's embedded payload. This triggers the execution of the malicious code, leading to remote code execution with elevated privileges.

M XDR solutions and behavioral monitoring can help detect and block malicious code execution during print jobs.

Source: evilsocket.net/2024/09/26/Attacking-UNIX-systems-via-CUPS-Part-I/

Footnotes

[1] Cups-browsed automatically discovers and configures network printers using protocols like DNS-SD or mDNS. It listens for printer advertisements and adds them to the local printer list without requiring manual configuration.

[2] Internet Printing Protocol (IPP) is a network protocol used for communication between client devices and printers. It allows users to send print jobs, query printer status, and manage print queues over a network.

[3] A PostScript Printer Description (PPD) file defines the characteristics and capabilities of a printer, such as supported paper sizes, resolution, and fonts. It is used by the operating system to ensure proper communication between the printer and print drivers during print jobs.

[4] The FoomaticRIPCommandLine is a parameter used in the Foomatic printing system, which allows arbitrary command execution for print job filtering. It processes print jobs by passing commands to a specific backend or filter, such as converting job formats or handling printer-specific functions.

The Fallacy of Default Trust

Many systems and services operate with default configurations, which are often assumed to be secure. This attack demonstrates the dangers of such assumptions. The CUPS browsing service allows unrestricted access in its default state, assuming that users will configure it securely, which is rarely done. This fallacy occurs when administrators rely too heavily on defaults, assuming they are well-protected out-of-the-box, when in reality, these defaults can expose the system to vulnerabilities.

CrowdStrike Faulty Update

The Fallacy of Automatic Updates

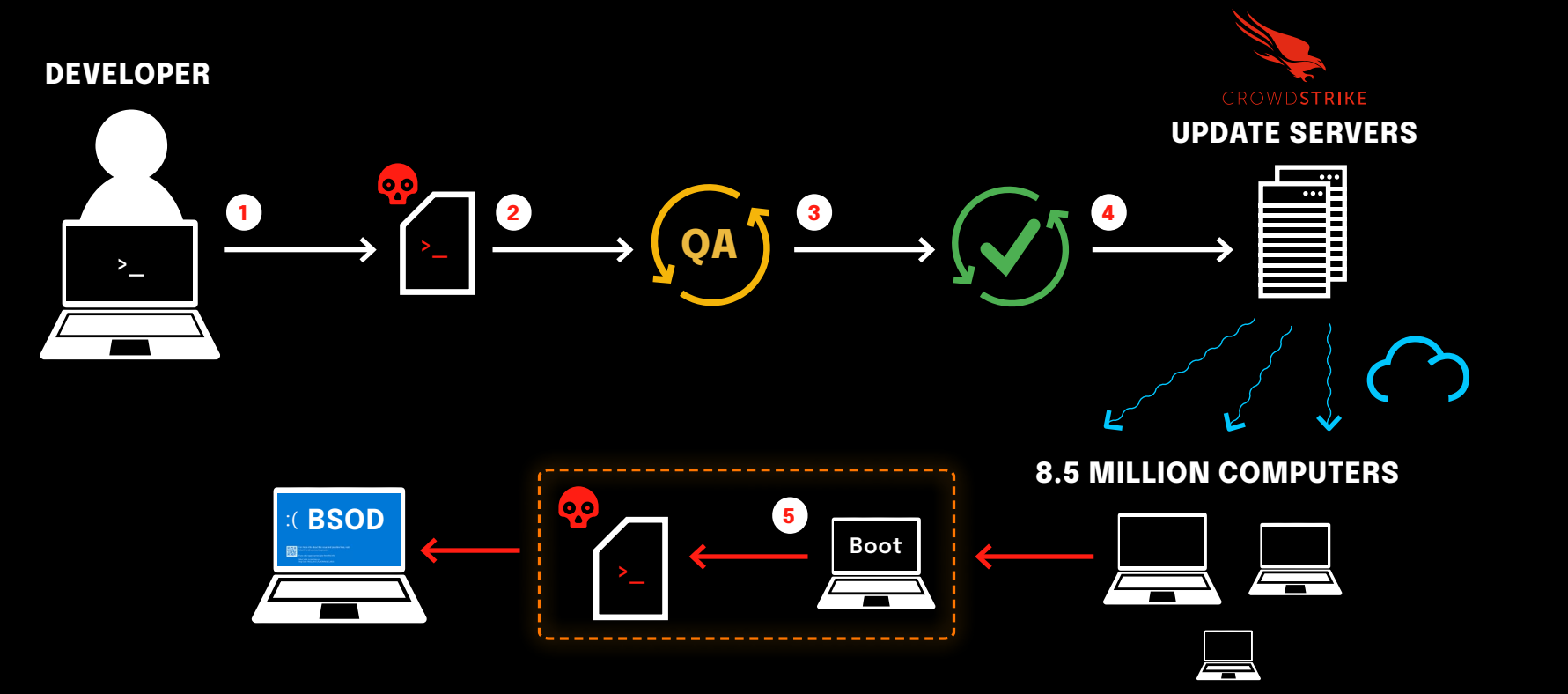
"We don't have a cybersecurity problem. We have a software quality problem"
Jen Easterly, US Cybersecurity & Infrastructure Security Agency director

CrowdStrike Falcon is an XDR platform which specializes in endpoint detection and remediation (EDR). On July 19, 2024, CrowdStrike released an automatic update for their Falcon product, which contained a defective file that impacted over 8.5 million systems and an estimated financial losses of at least $10 billion globally. Due to a misconfiguration in the Quality Assurance (QA) process, the faulty file went undetected, leading to widespread system crashes and Blue Screen of Death (BSOD) errors across various sectors, including hospitals, airports, banks and government agencies. This incident highlights the risks inherent in automatic updates, especially for critical systems, where even minor oversights can result in significant operational disruptions.

mSOC confidence score	Confirmed
Threat category	Security Patches/Updates
Severity	Critical

ATT&CK Technique	Attack Strategy	Evasion	Complexity	Target Type
T1072 - Software Deployment Tools	Faulty update appears as a legitimate software patch	Use of legitimate update channels	High	Any

ATT&CK Mitigation	Attack vector	Detection	Threat level	Threat Actor Type
M1018 - Application Isolation and Sandboxing	Defective update file	Testing updates in isolated environments	Medium	Any



Update Creation

1. CrowdStrike developers created an update for the Falcon product, which unfortunately included a defective file that would later cause issues. This file contained a logic error that would cause system crashes when processed.

Quality Assurance (QA) Process

2. The update was sent to the Quality Assurance team for testing. However, due to a misconfiguration in the QA checks, the defective file was not detected.

Approval for Release

3. Since the quality and assurance (QA) process did not catch the defective file, the update was marked as valid and staged for release.

Automatic Distribution

4. The update was released on July 19, 2024, 04:09 UTC and then automatically distributed to all systems running CrowdStrike's Falcon product, impacting over 8.5 million devices globally.

System Failures

5. Upon installation, the defective file caused systems to crash, leading to Blue Screen of Death (BSOD) errors. This BSOD was triggered due an out-of-bound memory read by the Windows sensor client. It is good to note that the update (Channel) files are not kernel drivers themselves, but are processed by Crowstrike's kernel-level code.

Widespread Impact

Although CrowdStrike quickly identified the issue and reverted the change at 05:27 UTC, roughly 1.5h after release the harm was already done. The incident left millions of systems unusable and underscored the risks associated with automatic updates, particularly for critical environments where even minor mistakes can cause severe operational disruptions.

The Fallacy of Automatic Updates

Automatic updates are generally beneficial, as they help protect systems against vulnerabilities and in the case of security solutions, often include new detection mechanisms for potential attacks. However, if an update contains defective files, it can lead to system issues or false positive detections. Although this is rare, for critical systems, it might be advantageous to consider a different update strategy, such as implementing a delay or review period before automatically installing updates.

XZ Utils Citical Backdoor

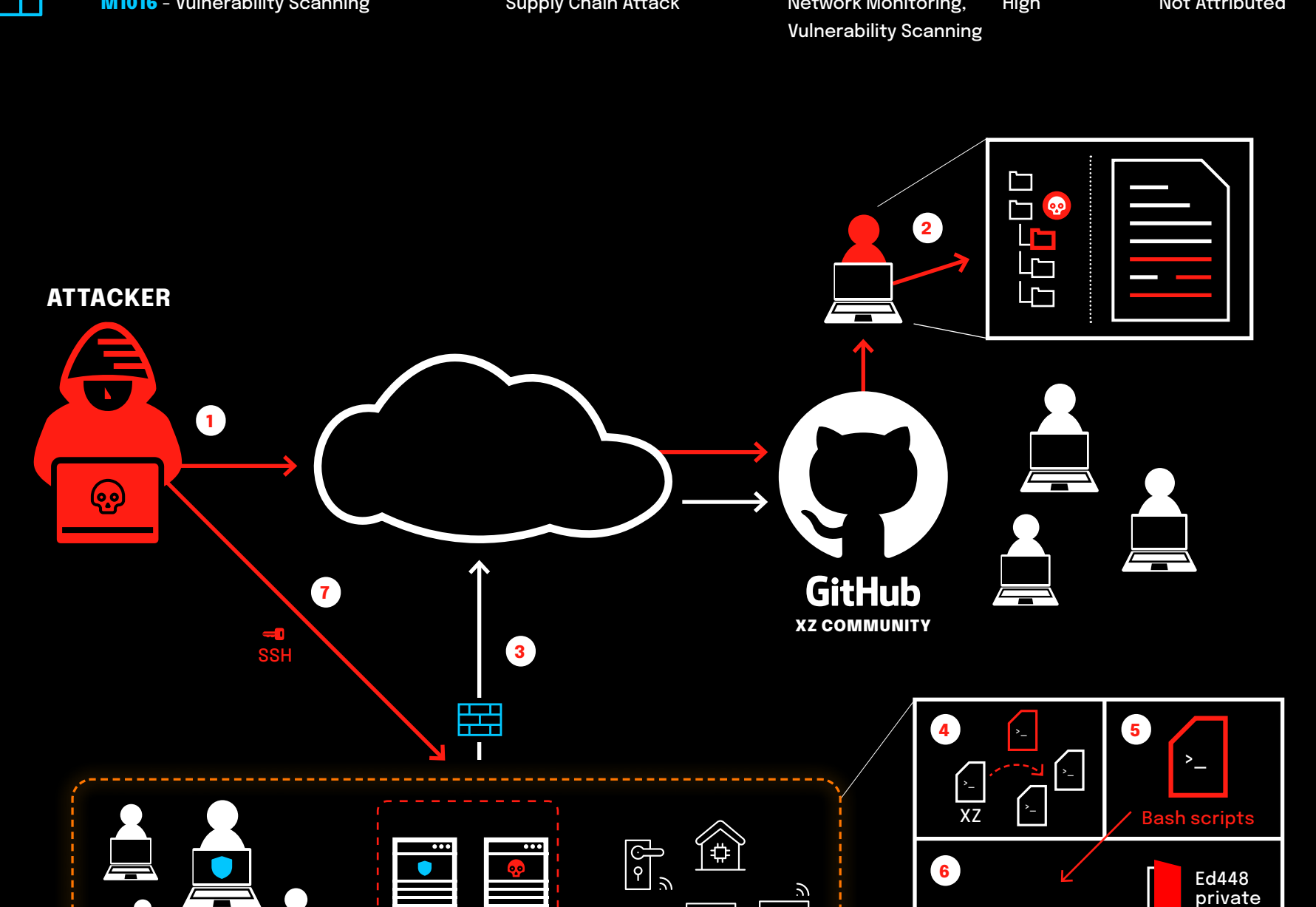
The Fallacy of Secure Open Source Code

On March 29, 2024, a critical backdoor was discovered in XZ Utils, specifically in versions 5.6.0 and 5.6.1. The backdoor exploits the widely-used data compression library liblzma, which integrates into Linux systems. The malicious code allows attackers with a specific Ed448 private key to gain unauthorized remote access and execute arbitrary code via SSH on vulnerable systems. This vulnerability was part of a supply chain attack and affects many Linux distributions.

mSOC confidence score	Confirmed
Threat category	Supply Chain Attack
Severity	Critical

ATT&CK Technique	Attack Strategy	Evasion	Complexity	Target Type
T1195 - Supply Chain Compromise	Social engineering to inject malicious code in open source project	Code Obfuscation	High	Any

ATT&CK Mitigation	Attack vector	Detection	Threat level	Threat Actor Type
M1016 - Vulnerability Scanning	Supply Chain Attack	Network Monitoring, Vulnerability Scanning	High	Not Attributed



Building credibility

1. The threat actor creates a fake GitHub account under the alias JIAT75. Over a period of three years, the attacker builds a credible reputation through multiple social engineering tactics, ultimately gaining trust within the open-source community. Leveraging these efforts, JIAT75 successfully joins the Tukaani Project–the maintainers of XZ Utils [1]–as a contributor through carefully crafted social engineering.

M Regular audits of contributor activity and vetting new contributors more rigorously could reduce the risk of social engineering attacks. Additionally, implementing a four-eyes principle before committing changes to popular projects can aid in detecting malicious code.

Malicious code injection

2. The threat actor introduces malicious code into the release artifacts of XZ Utils versions 5.6.0 and 5.6.1. This is achieved through advanced obfuscation techniques, allowing the code to bypass regular reviews and be distributed undetected in the official release.

M Enforce code review policies and use automated static code analysis tools to detect obfuscated or suspicious code patterns.

The infection spreads

3. During routine updates, victims unknowingly download the compromised XZ Utils package. This infects their devices by pulling the malicious code directly from the repository.

M Encouraging users and organizations to prioritize long-term support (LTS) stable versions over bleeding-edge releases can help limit the risk of exposure to vulnerabilities introduced in less-tested versions.

Let the trojan horse in

4. Upon installation of the compromised package, the liblzma [2] library extracts a prebuilt object file disguised within a test file. This object file is then used to modify specific functions in liblzma, enabling the malicious payload.

Malicious payload & script execution

5. A series of obfuscated bash scripts are created and executed on the victim's machine. These scripts carry out operations such as decompressing the payload and executing additional commands, further embedding the backdoor.

Backdoor in place

6. The malicious code utilizes OpenSSL to perform cryptographic operations involving a hardcoded Ed448 private key [3]. This allows the attacker to decrypt the payload and bypass SSH authentication mechanisms, enabling unauthorized access using a custom key.

SSH connection

7. At this stage, the attacker can connect to the infected system via SSH, using the corresponding private key to authenticate and gain full control over the victim's machine.

M Restricting inbound SSH connections using firewall rules to only accept connections from trusted IP addresses significantly reduces the attack surface, limiting the potential for unauthorized access. If it not possible to restrict the SSH connectivity because of any reason, then port knocking [4] could be implemented in order to mitigate this risk.

Footnotes

[1] XZ Utils is a free, open-source data compression software that uses the LZMA2 compression algorithm, known for its high compression ratio and efficiency. It is widely used in Linux and Unix-like systems for compressing and decompressing large files, especially in packaging and software distribution.

[2] Liblzma is the core library of XZ Utils, responsible for implementing the LZMA2 compression algorithm. It handles data compression and decompression tasks and is used in various applications to reduce file sizes efficiently.

[3] An Ed448 private key is part of the Ed448 elliptic curve cryptography system, which provides high security for digital signatures and encryption. It is used to sign messages and authenticate securely, offering stronger protection than many other elliptic curves due to its 448-bit key size.

[4] The primary purpose of port knocking is to prevent an attacker from scanning a system for potentially exploitable services by doing a port scan, because unless the attacker sends the correct knock sequence, the protected ports will appear closed.

The Fallacy of Secure Open Source Code

Many people assume that open-source software is inherently secure because the code is open to review by the entire community. This belief overlooks the fact that even with open access, not all code is thoroughly audited, and malicious contributors can still insert backdoors or vulnerabilities. The XZ Utils backdoor incident demonstrates the fallacy of trusting open-source projects simply because their code is visible. Without rigorous vetting of contributors, automated security tools, and ongoing audits, even widely used open-source software can be compromised, exposing systems to significant risks.

Taxonomy

ATT&CK Technique
Which technique of the MITRE ATT&CK framework does the threat correspond to.

ATT&CK Mitigation
Which mitigation of the MITRE ATT&CK framework can be applied.

Attack Strategy
Plan devised by the attacker to exploit specific system vulnerabilities.

Attack Vector
What is the primary method of attack.

Evasion
Tactics used by the attacker to avoid detection or bypass security.

Detection
Mechanism to identify malicious activities or system anomalies.

Complexity
How easy it is to exploit the vulnerability or carry out the attack.

Threat Level
How severe the threat is.

Target Type
The category of organization that may potentially be targeted.

Threat Actor Type
What type of threat actor may be involved.

CONTACT US

mSOC score explanation

We assign scores to both our sources and the news items. Sources are scored on a numeric scale ranging from 0 (untrustworthy) to 5 (verified), while news items are scored with a letter, ranging from E (unreliable) to A (reliable). By considering the scores of both the source and the news item and the quality of the available information, we classify the overall reliability into three categories: Confirmed, Verified, and Credible. Interested in learning more about our reliability scoring system for sources and news items? Our Threat Intelligence team would be happy to walk you through our procedure, so please don't hesitate to reach out.

threat-talks.com

ON2IT ZERO TRUST INNOVATORS — amsix