

# Investigating **XZ/LIBLZMA**

for SOC Teams



# TABLE OF CONTENTS

**01**

Alert

**04**

Detection

**07**

Analysis

**14**

Summary

**15**

Lesson Learned

# Alert

Based on the information that the alert provided, it appears that there is a suspicious file detected on a Linux server named "SSHDevServer01" with an IP address of 172.16.17.121. The Alert is triggered by the SOC271 rule for XZ/LIBLZMA Backdoor Implant Detected CVE-2024-3094.

CVE-2024-1709 is an authentication bypass vulnerability that allows attackers to create system admin accounts on vulnerable instances and use them for their own malicious ends.

The Level 1 (L1) analyst identified that the host is accessible from the internet and detected a file named liblzma.so.5.6.1 as suspicious according to VirusTotal, in relation to CVE-2024-3094. Additionally, the L1 analyst's analysis revealed no suspicious network traffic. The detected file is located at '/usr/local/lib/liblzma.so.5.6.1'.

SEVERITY	DATE	RULE NAME	EVENTID	TYPE	ACTION
High	2024-04-04 4:52	★ SOC271 - XZ/LIBLZMA Backdoor Implant Detected CVE-2024-3094	247	Malware	
<b>Event Details</b>					
EventID :	247				
Event Time :	2024-04-04 4:52				
Rule :	SOC271 - XZ/LIBLZMA Backdoor Implant Detected CVE-2024-3094				
Level :	Incident Responder				
Hostname :	SSHDevServer01				
Ip Address :	172.16.17.121				
File Name :	liblzma.so.5.6.1				
File Path :	/usr/local/lib/				
File Hash :	9b368d0ad8b3bda5eabfdf8a40944f4dd270955bab868da9a51beedcfde38699				
Trigger Reason :	Suspicious Hash detected for the file liblzma.so.5.6.1				
Device Action :	Allowed				
L1 Note :	This server, SSHDevServer01, hosts an OpenSSH server that is exposed to the internet for development team testing. I am escalating this for your review due to a detected suspicious hash for the file "liblzma.so.5.6.1". While my initial analysis did not uncover any suspicious traffic, a check on VirusTotal flagged the file hash as malicious, raising concerns about a potential compromise. Please also investigate to determine if the host is affected by the CVE-2024-3094 vulnerability.				

The device action is marked as "allowed", indicating that no action was taken by the device to prevent or block the file.

Based on the provided trigger reason, a suspicious hash associated with the file liblzma.so.5.6.1 is detected. The file hash is:

9b368d0ad8b3bda5eabfdf8a40944f4dd270955bab868da9a51beedcfde38699

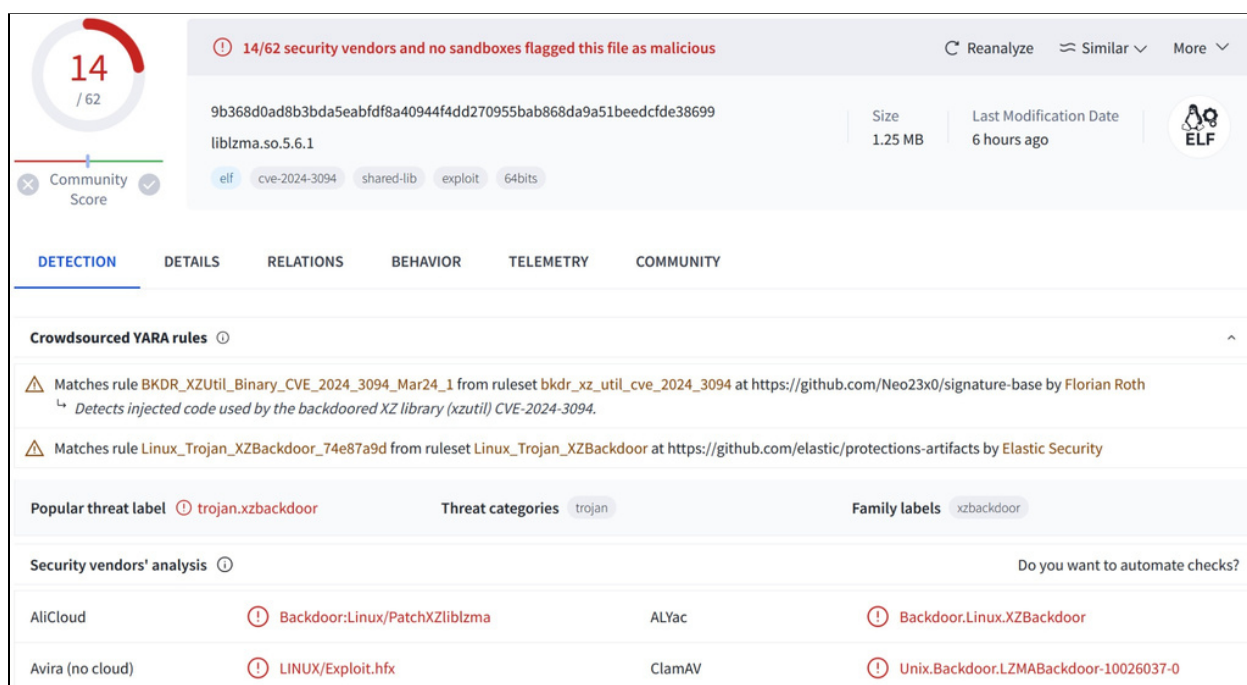
Overall, the system shows signs of potential malicious activity. Further investigation is required to assess the scope of this activity and determine appropriate remedial actions.

# Detection

## Verify

As a security analyst, one of the first steps we take to verify the alert and determine whether it is a false positive or a true positive incident is to analyze the logs collected from the host by our security products.


The first step we can take to investigate the hash value of the suspicious file is to use online threat intelligence platforms such as VirusTotal, Hybrid Analysis and MalwareBazaar.



The screenshot shows the VirusTotal interface for a file. At the top, a red circle indicates that 14 out of 62 security vendors have flagged the file as malicious. The file's SHA-256 hash is 9b368d0ad8b3bda5eabfdf8a40944f4dd270955bab868da9a51beedcfe38699, and it is identified as liblzma.so.5.6.1. The file size is 1.25 MB and it was last modified 6 hours ago. The file type is ELF. The analysis shows that the file matches two YARA rules: BKDR\_XZUtil\_Binary\_CVE\_2024\_3094\_Mar24\_1 and Linux\_Trojan\_XZBackdoor\_74e87a9d. The popular threat label is trojan.xzbackdoor, and the family label is xzbackdoor. The security vendors' analysis shows that the file is flagged as malicious by AliCloud, Avira (no cloud), and ClamAV.

Security vendors' analysis	Do you want to automate checks?
AliCloud	Backdoor:Linux/PatchXZliblzma
Avira (no cloud)	LINUX/Exploit.hfx
ClamAV	Unix.Backdoor.LZMABackdoor-10026037-0

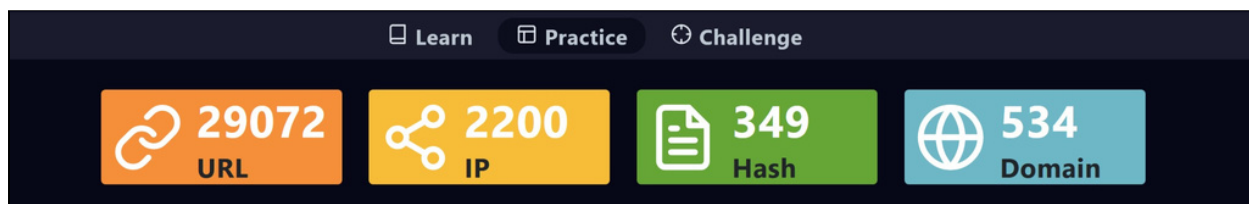
Based on the information provided by VirusTotal, it appears that the file is malicious and has been flagged as malicious by 14 out of 62 security vendors. The file has been labeled as "TROJAN" and the family Label is "xzbackdoor".



The summary bar shows the popular threat label as trojan.xzbackdoor, the threat categories as trojan, and the family labels as xzbackdoor.

In the tags, we also see that the file is associated with the [cve-2024-3094](#)

On the LetsDefend threat intel tab, you'll find a comprehensive database dedicated to cataloging maliciously used information, such as IP addresses, domains, and other indicators of compromise.



<https://app.letsdefend.io/threat-intelligence-feed>

Upon cross-referencing the destination IP address discovered in the log management system with the Threat Intel tab, it was determined that the file has been tagged as both backdoor and Linux.xzBackdoor.

The screenshot shows the search interface with filters for 'Free text search', 'Date range', 'Search by data type' (set to 'Hash'), 'Search by data' (set to 'a9a51beedcfd38699'), and 'Search by tag'. A 'Search' button is visible. Below the search bar, a table displays the results.

DATE	DATA TYPE	DATA	TAG	DATA SOURCE
Apr, 05, 2024, 09:16 AM	Hash	9b368d0ad8b3bda5eabdf8a40944f4dd270955bab...	Backdoor.Linux...	Anonymous

Our analysis confirmed that the alert is a true positive (TP), and the malicious file has been seen on the host. This incident warrants further investigation and an appropriate response.

## CVE-2024-3094

After a thorough analysis, we have verified that the file "liblzma.so.5.6.1" being detected as suspicious by VirusTotal in the context of CVE-2024-3094 is a true positive (TP). This implies that the malicious file is present on the host and requires immediate attention.

The primary goal now is to delve deeper into the specifics of CVE-2024-3094. Understanding the nature, impact, and potential exploits associated with this CVE will aid in devising an effective remediation strategy.

# XZ BACKDOOR: CVE-2024-3094



On Friday, March 29, after investigating anomalous behavior in his Debian sid environment, developer [Andres Freund](#) contacted an [open-source security mailing list](#) to share that he had discovered an upstream backdoor in the widely used command line tool XZ Utils (liblzma). The backdoor, added by an open-source committer who had been working on the tool for several years, affects XZ Utils versions 5.6.0 and 5.6.1. It has been assigned [CVE-2024-3094](#).



**SANS.edu Internet Storm Center**  
@sans\_isc

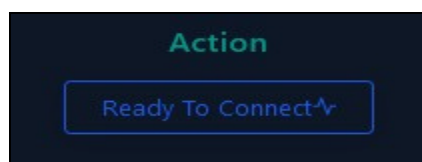
...

A quick note about xz-utils backdoor:  
1 - luckily, this was caught early.  
2 - most run xz-utils 5.2/5.4. 5.6 is bad.  
3 - quick check: `xz -V`  
4 - Thanks to people who paid attention

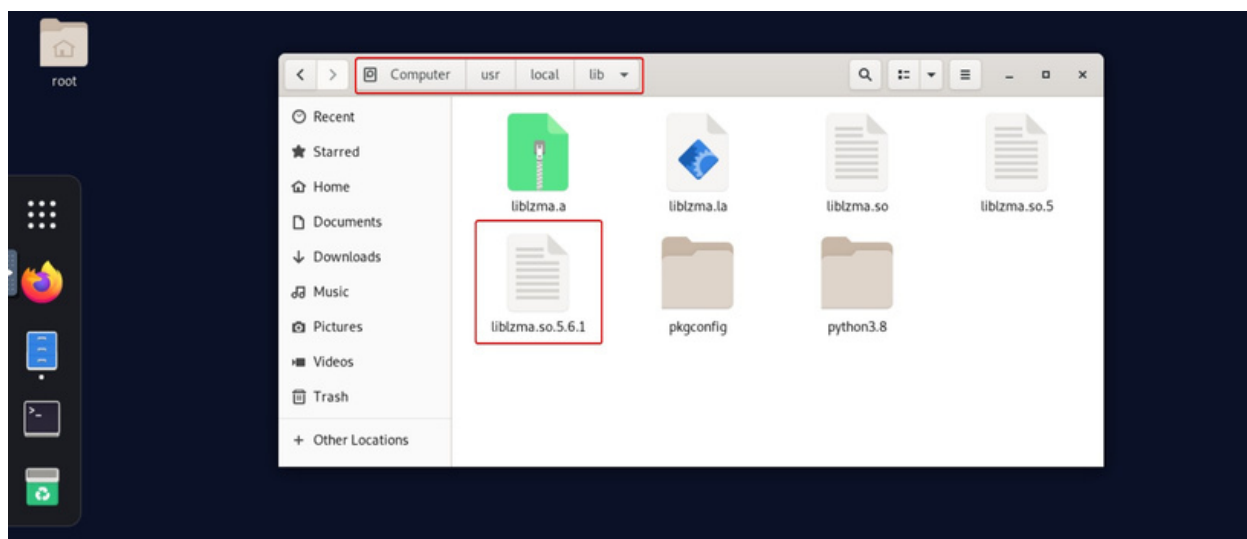
[openwall.com/lists/oss-secu...](https://openwall.com/lists/oss-secu...)

# Analysis

*We can proceed with connecting to the host machine for further analysis. This can easily be done from the Endpoint Security tab by searching for the hostname or IP address and clicking the "Connect" button.*



We have identified that the suspicious file is located in the /usr/local/lib folder in alert details. When we checked the related path, we found that the file still exists.



We can use "sha256sum" command to get the hash of the file and crosscheck it with the hash from alert details.

```
root@ip-172-31-26-216: /usr/local/lib# ls
liblzma.a  liblzma.la  liblzma.so  liblzma.so.5  liblzma.so.5.6.1  pkgconfig  python3.8
root@ip-172-31-26-216: /usr/local/lib# sha256sum liblzma.so.5.6.1
9b368d0ad8b3bda5eabfdf8a40944f4dd270955bab868da9a51beedcfde38699  liblzma.so.5.6.1
root@ip-172-31-26-216: /usr/local/lib#
```

We have identified the file successfully and verified it by cross-checking hashes.

Based on the information from the email and other published sources, it has been discovered that the XZ Utils 5.6.0 and 5.6.1 release tarballs contain a backdoor.

### Facts

- CVE-2024-3094
- XZ Utils 5.6.0 and 5.6.1 release tarballs contain a backdoor. These tarballs were created and signed by *Jia Tan*.
- Tarballs created by Jia Tan were signed by him. Any tarballs signed by me were created by me.

<https://tukaani.org/xz-backdoor/>

To begin our assessment, the first step is to determine the version of our XZ Utils installations. Using the command “xz -V” is a quick and straightforward method to check the installed version.

```
root@ip-172-31-26-216: /usr/local/lib# xz -V
xz (XZ Utils) 5.6.1
liblzma 5.6.1
root@ip-172-31-26-216: /usr/local/lib#
```

The version of xzutils installed on the SSHDevServer01 host is 5.6.1, suggesting that the machine may have been infected with the backdoor. Another ways to check if that the host is affected by the xz backdoor is using the script “[detect.sh](#)” given in the mail of Andres Freund.

Andres Freund

View attachment "[injected.txt](#)" of type "text/plain" (8236 bytes)

Download attachment "[liblzma\\_la-crc64-fast.o.gz](#)" of type "application/gzip" (36487 bytes)

Download attachment "[detect.sh](#)" of type "application/x-sh" (426 bytes)



The code itself is looks like this.

```
GNU nano 4.8 detect.sh Modified
#!/bin/bash

set -eu

# find path to liblzma used by sshd
path="$(ldd $(which sshd) | grep liblzma | grep -o '^[^ ]*')"
```

```
# does it even exist?
if [ "$path" == "" ]
then
    echo probably not vulnerable
    exit
fi

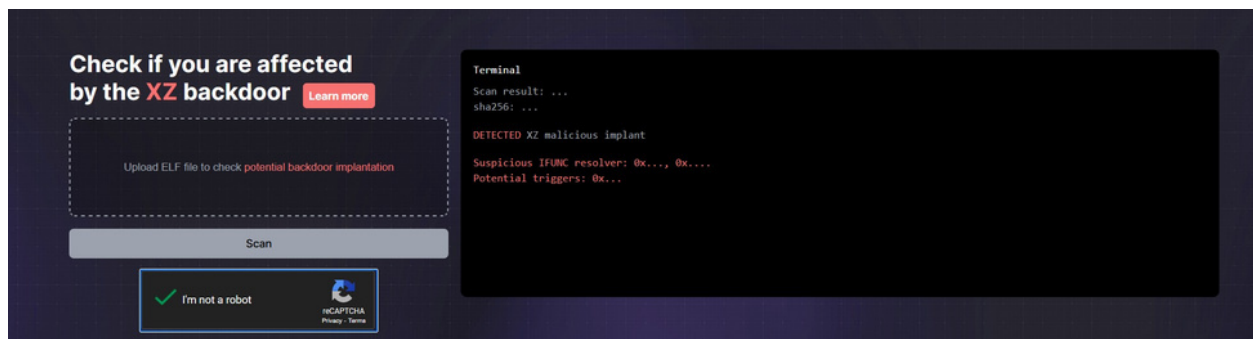
# check for function signature
if hexdump -ve '1/1 "%.2x"' "$path" | grep -q f30f1efa554889f54c89ce5389fb81e7000000804883ec28488
then
    echo probably vulnerable
else
    echo probably not vulnerable
fi
```

Get Help Write Out Where Is Cut Text Justify Cur Pos  
Exit Read File Replace Paste Text To Spell Go To Line

By running the script we also see that the host is probably vulnerable to CVE-2024-3094

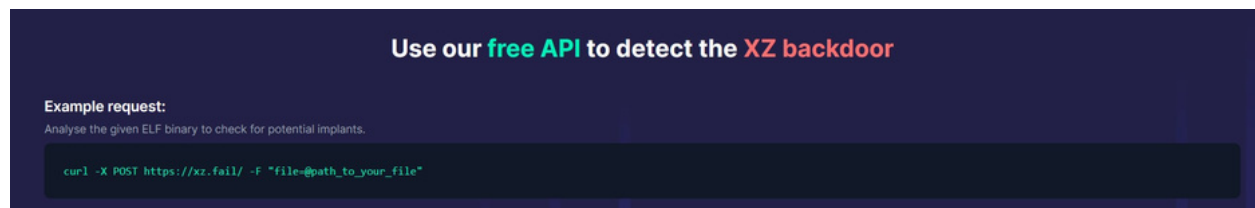
```
root@ip-172-31-26-216:~# ./detect.sh
probably vulnerable
root@ip-172-31-26-216:~#
```

There is also a website named xz.fail which will also checks if the system is affected by the XZ backdoor.

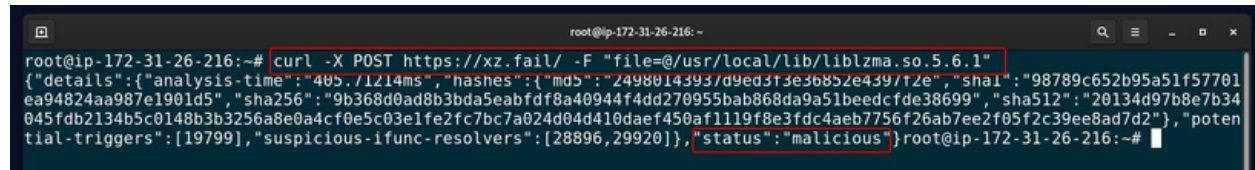


[Checking for Backdoor Implant](#)

We can use the xz.fail API services to detect the XZ backdoor.

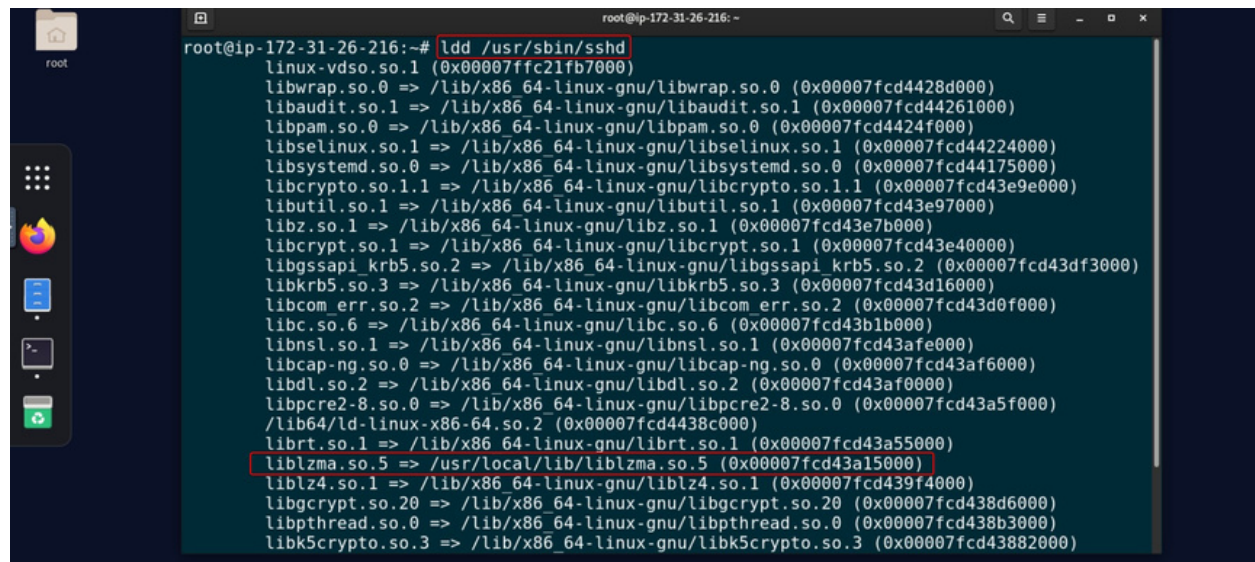


`curl -X POST https://xz.fail/ -F "file=@path_to_your_file"`



Thanks to Andres Freund for his excellent work and detailed report, and to Vegard Nossum for the detect.sh script, we discovered that our system is affected by CVE-2024-3094.

The ldd command is used to print the shared library dependencies of an executable or shared library on Linux systems. When you run ldd on an executable or a shared library, it shows you the shared libraries required by that binary or library. When you run this command, it will list the shared libraries that sshd depends on.



Based on the information provided, the image indicates that sshd is dependent on liblzma.so.5. When sshd (the SSH daemon) performs public key authentication, it calls the RSA\_public\_decrypt function. Due to the manipulation in the Makefile, this call redirects to the attacker's code.

## Initial Access

The presence of malicious files on the computer should make us think about initial access. It is crucial to investigate the initial access point of the attacker in order to determine how they were able to gain unauthorized access to the system.

The initial access method employed in this sophisticated attack was a Supply Chain Compromise. The attacker successfully infiltrated the upstream xz repository and compromised the xz tarballs. This infiltration led to the introduction of a malicious version of the library, specifically targeting liblzma.so.5.6.1.

The malicious activity was primarily present in the distributed tarballs of xz releases 5.6.0 and 5.6.1. A specific line in the tarball's code, which is absent from the original upstream source, was identified as the injection point for the backdoor. This injected line initiates an obfuscated script during the configure phase. The script, when executed, modifies the Makefile in the liblzma directory.

Further inspection of the compromised repository revealed that the malicious code was embedded in obfuscated form within certain test files. These files were introduced in the xz 5.6.0 release, yet they were not actively utilized for any tests. Notably, these files were subsequently adjusted in the 5.6.1 release to address issues that arose due to the injected code.

Upon further investigation, it was discovered that the initial access was achieved through a Supply Chain Compromise which enabled the attacker to implant the backdoor to the host machine.

Supply Chain Compromise - T1195
---------------------------------

Based on the analysis conducted thus far, no indications of compromise (IOCs) have been identified on other machines within the network. This suggests that the scope of the incident is limited to the specific machine under investigation.

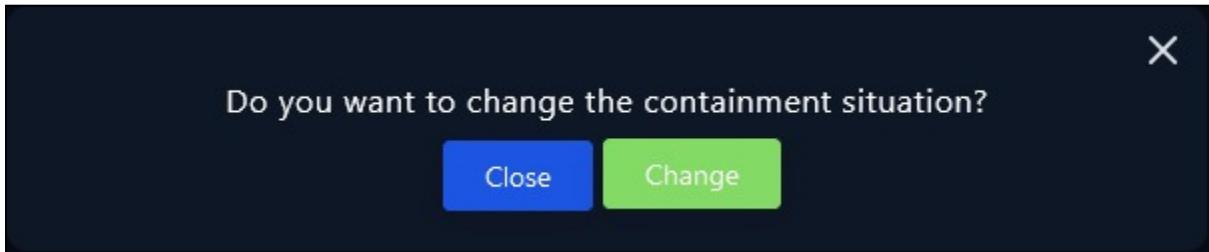
## Execution

Upon thorough investigation, it has been determined that the malware has not been executed on the affected device. The compromise originated from the compromised xz repository, which resulted in the host being equipped with the backdoored version 5.6.1 of xz-utils.

Despite the installation of the backdoored software, our analysis did not identify any signs of external traffic or Remote Code Execution (RCE) associated with the sshd process. This absence of malicious activity suggests that while the backdoor has been successfully installed, it has not yet been activated or executed on the device.

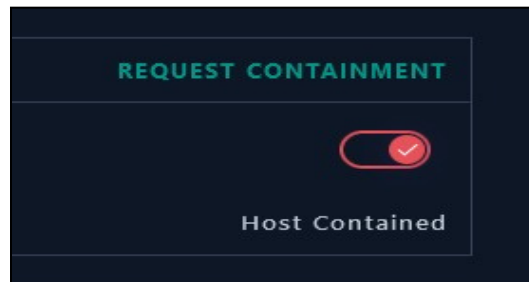
# Containment

Based on the information gathered during the investigation, it is highly likely that the system has been compromised. To prevent further data loss or unauthorized access, it is recommended to isolate the system from the network immediately.



Isolation of the host can be made from the endpoint security tab.

Hostname	SSHDevServer01
IP Address	172.16.17.121



## Summary

This incident response report documents a security alert triggered by the SOC271 rule for XZ/LIBLZMA Backdoor Implant Detected CVE-2024-3094 on a Linux system named "SSHDevServer01" with the IP address 172.16.17.121. The alert was associated with a suspicious hashed "liblzma.so.5.6.1" which was detected on the device. Upon investigation, it was confirmed that the file was associated with the CVE-2024-3094.

Analysis of the file's hash on VirusTotal indicated that it was flagged as malicious by 14 out of 62 security vendors and labeled as a "TROJAN and Backdoor.Linux.XZBackdoor." Further investigation revealed that the file is backdoored, confirming it as a true positive incident.

The report then delves into the initial access point, which was determined to be a Supply Chain Compromise attempt. Suspicious outbound connections were not observed from the infected host machine during the alert time.

The host is vulnerable to CVE-2024-3094 due to the infected liblzma, yet no malicious connections or executions have been detected. This suggests the system is infected but not yet fully compromised.

The incident appears to be limited to the specific machine under investigation, and further remediation actions are required to address the security breach and prevent future incidents.

## Lesson Learned

- It is important to keep all software up-to-date to reduce the risk of being vulnerable to known or unknown exploits
- Trusting third-party repositories without proper validation and verification can introduce vulnerabilities into the system. Organizations must prioritize the vetting of third-party software and regularly monitor them for any signs of compromise.
- reminder of the risks associated with supply chain attacks, where attackers compromise a part of the software supply chain to inject malicious code into legitimate software.
- Regularly monitor and analyze process trees to identify any unusual or suspicious parent-child relationships, which can provide insights into the execution techniques employed by attackers.

## Remediation Actions

- Downgrading(orupgrading)XZUtilpackagestoasecureversionbasedonthe relevant advisory
- BlockingexternalSSHaccess.
- Isolatethecompromisedmachinefromthenetworktopreventtheattackerfrom accessing other resources and systems within the organization.

# Appendix

## MITRE ATT&CK

Initial Access
T1195: Supply Chain Compromise
T1195.003: Compromise Hardware Supply Chain
T1195.001: Compromise Software Dependencies and Development Tools
T1195.002: Compromise Software Supply Chain

MITRE Tactics	MITRE Techniques
Initial Access	T1195 Supply Chain Compromise

## Artifacts

Filename	SHA256 Value - Path
liblzma.so.5.6.1	9b368d0ad8b3bda5eabfdf8a40944f4dd270955bab868da9a51beedcfde38699