deepwatch

August 2022

INCIDENT INTEL REPORTS

Novel Backdoor Discovered

XMRig Technical Analysis: Threat Actor Leverages Confluence Vulnerability to Deploy Novel Backdoor

Part 3

Category: Incident Intel Reports

Headlines: Part 3 XMRig Technical Analysis: Threat Actor Leverages Confluence Vulnerability to Deploy Novel Backdoor

Author: @r1n9w0rm

Overview

Deepwatch has observed threat actors exploiting out-of-date versions of Atlassian Confluence Server and Data Center, leading to the installation of the XMRig crypto-miner.

Vulnerability Details

As detailed in Part 1 of this report, the suspected vulnerability used in this attack was CVE-2022-26134, which affects out-of-date versions of Confluence Server and Data Center, and allows remote code execution (RCE) under the privileges of the user running the service.

Affected Products

For all affected versions and products, see the security advisory published by Atlassian at:
https://confluence.atlassian.com/doc/confluence-security-advisory-2022-06-02-1130377146.html

Technical Analysis

XMRig is a legitimate open source cryptocurrency miner for mining the Monero (XMR) cryptocurrency. It is also a popular choice amongst threat actors that use it to mine the cryptocurrency on hijacked systems. After extracting the user account for this particular miner, we have discovered that the Threat Actor(s) who control it may have received more than 652 XMR (worth $82,176 at this time) in rewards for mining on hijacked systems. We have not found any evidence in this instance that the Threat Actor(s) successfully executed the miner, rather we only found evidence that it was present in an artifact from the system. Within OBJECTS.DATA, we identified a base64 encoded PE file. OBECTS.DATA stores WMI object data and is located on disk at C:\WINDOWS\system32\wbem\Repository\OBJECTS.DATA. Once decoded from base64, we found it is a packed loader that uses process hollowing to execute the XMRig crypto-miner.

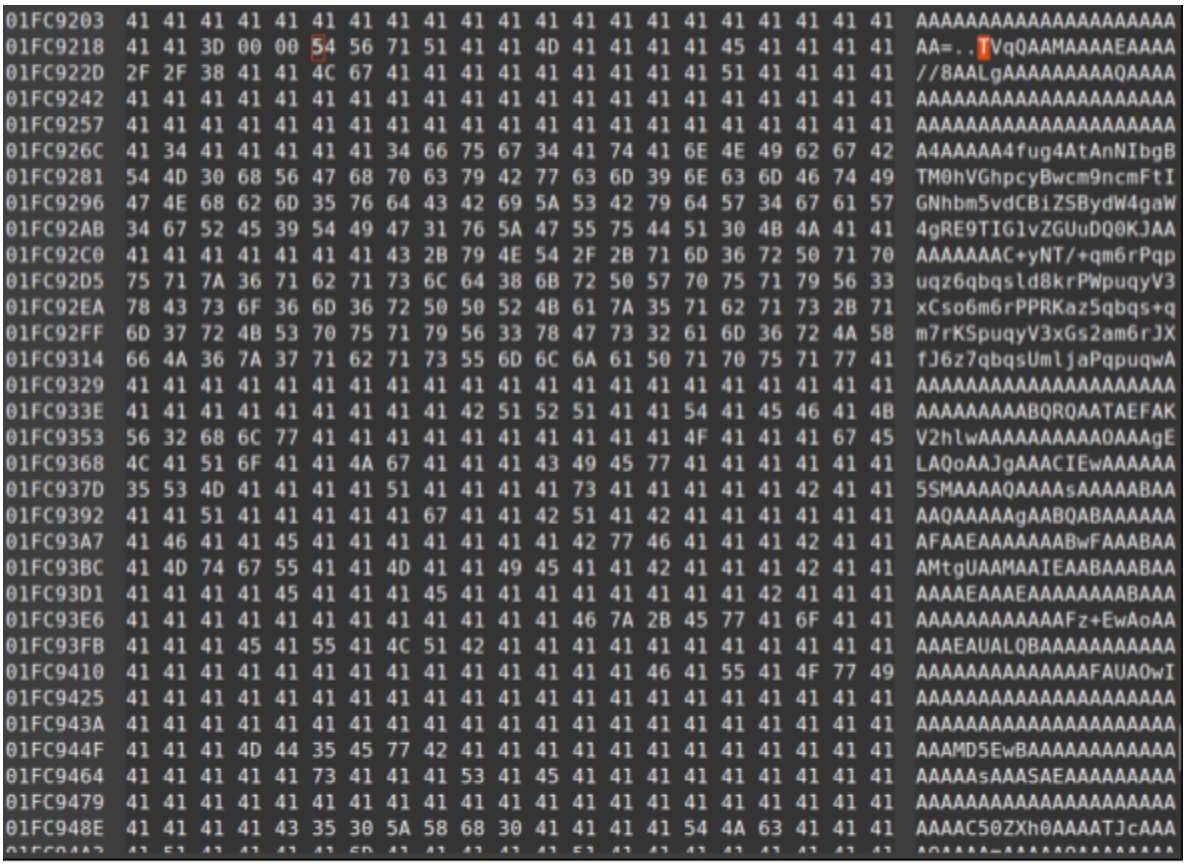

Figure 1 - PE within OBJECTS.DATA

Following the main routine, a subroutine makes use of the process hollowing technique on the schtasks (legitimate Windows system utility located at C:\Windows\System32\schtasks.exe), in order to execute XMRig within the virtual address space of schtasks. This technique is documented in the Mitre Att&ck framework as technique ID T1055.012. First the schtasks.exe process is started with the CREATE_SUSPENDED flag as seen below.

```
mov     [ebp+StartupInfo.hStdError], eax
lea     eax, [ebp+StartupInfo]
push    eax             ; lpStartupInfo
push    esi             ; lpCurrentDirectory
push    esi             ; lpEnvironment
push    4               ; dwCreationFlags -> CREATE_SUSPENDED
push    1               ; bInheritHandles
push    esi             ; lpThreadAttributes
push    esi             ; lpProcessAttributes
push    [ebp+lpCommandLine] ; lpCommandLine
mov     [ebp+StartupInfo.dwFlags], 101h
push    esi             ; lpApplicationName
call    ds:CreateProcessA
test    eax, eax
jz      short loc_4011AC
```

Figure 2 - CreateProcessA w/ suspended flag

Next, the Windows API method ZwUnmapViewOfSection is called in order to unmap the existing and legitimate schtask code. This is followed up by a call to the Windows API method VirtualAllocEx to allocate RWE (Read, Write, Execute) memory in the schtasks process.

Figure 3 - ZwUnmapViewOfSection

This is followed by calls to the Windows API method WriteProcessMemory, in order to write the XMRig payload to the RWE memory space. Finally, the Windows API methods SetThreadContext and ResumeThread start the execution of XMRig.

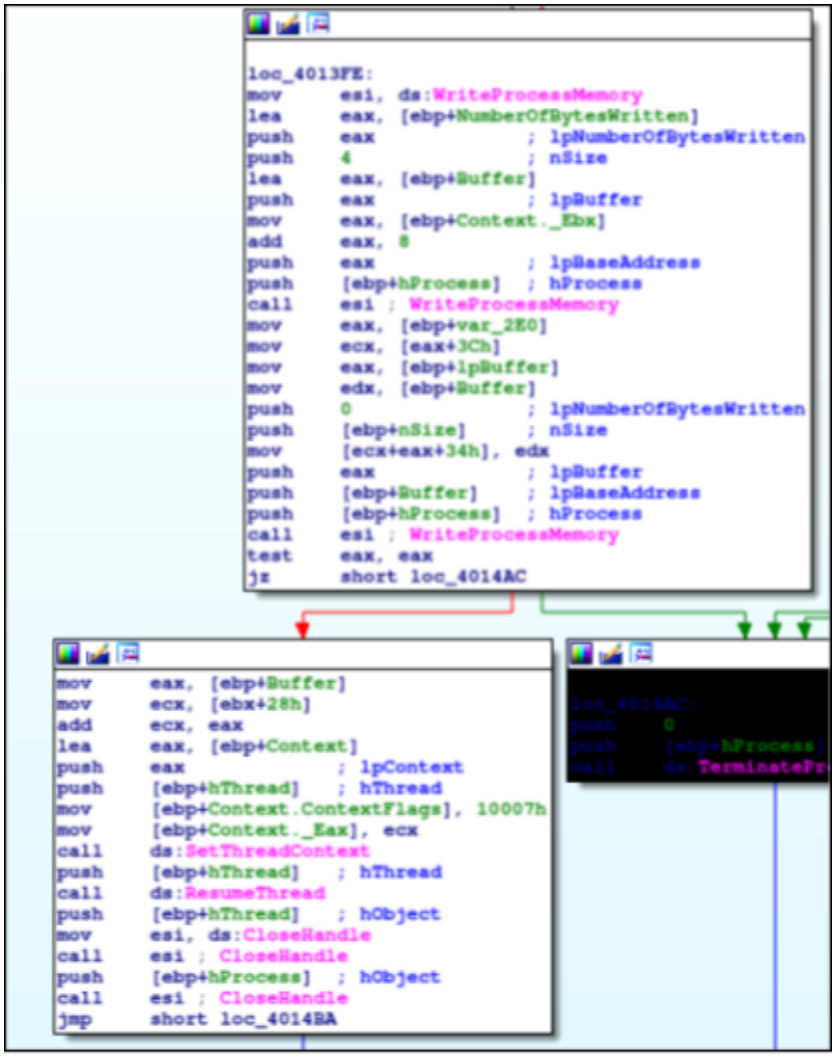

Figure 4 - Write/execute XMRig

When viewing the process list with a tool like task manager, we observed very high CPU usage for schtasks.exe, as seen in the figure below. This is highly suspicious, considering that schtasks is a simple command line utility for listing/creating/deleting scheduled tasks.

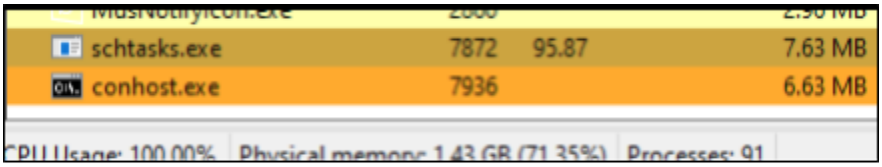

Figure 5 - High CPU usage

Upon analyzing the XMRig payload, we can see the user and mining pool URLs in the main subroutine.

```
mov     [ebp+var_B8], offset aStratumTcpXmrE ; "stratum+tcp://xmr-eu1.nanopool.org:1444"..
mov     [ebp+var_B4], offset aU ; "-u"
mov     [ebp+var_B0], offset a49wzduvq1dfwg3 ; "49WZduVQ1DFWG3scZxFT8hBY1JsoYuJVqMRe8UA"..
mov     [ebp+var_AC], offset aP ; "-p"
mov     [ebp+var_A8], offset asc_51F918 ; "x"
mov     [ebp+var_A4], offset aO_0 ; "-o"
mov     [ebp+var_A0], offset aStratumTcpXmrE_0 ; "stratum+tcp://xmr-eu2.nanopool.org:1444"
mov     [ebp+var_9C], offset aU_0 ; "-u"
mov     [ebp+var_98], offset a49wzduvq1dfwg3_0 ; "49WZduVQ1DFWG3scZxFT8hBY1JsoYuJVqMRe8UA"
mov     [ebp+var_94], offset aP_2 ; "-p"
mov     [ebp+var_90], offset asc_51F9B4 ; "x"
mov     [ebp+var_8C], offset aO_1 ; "-o"
mov     [ebp+var_88], offset aStratumTcpXmrU ; "stratum+tcp://xmr-us-east1.nanopool.org"..
mov     [ebp+var_84], offset aU_1 ; "-u"
mov     [ebp+var_80], offset a49wzduvq1dfwg3_1 ; "49WZduVQ1DFWG3scZxFT8hBY1JsoYuJVqMRe8UA"
mov     [ebp+var_7C], offset aP_3 ; "-p"
mov     [ebp+var_78], offset asc_51FA54 ; "x"
mov     [ebp+var_74], offset aO_2 ; "-o"
mov     [ebp+var_70], offset aStratumTcpXmrU_0 ; "stratum+tcp://xmr-us-west1.nanopool.org"
mov     [ebp+var_6C], offset aU_2 ; "-u"
mov     [ebp+var_68], offset a49wzduvq1dfwg3_2 ; "49WZduVQ1DFWG3scZxFT8hBY1JsoYuJVqMRe8UA"
mov     [ebp+var_64], offset aP_4 ; "-p"
mov     [ebp+var_60], offset asc_51FAF4 ; "x"
mov     [ebp+var_5C], offset aO_3 ; "-o"
mov     [ebp+var_58], offset aStratumTcpXmrA ; "stratum+tcp://xmr-asia1.nanopool.org:14"..
mov     [ebp+var_54], offset aU_3 ; "-u"
mov     [ebp+var_50], offset a49wzduvq1dfwg3_3 ; "49WZduVQ1DFWG3scZxFT8hBY1JsoYuJVqMRe8UA"
mov     [ebp+var_4C], offset aP_0 ; "-p"
mov     [ebp+var_48], offset asc_51FB90 ; "x"
mov     [ebp+var_44], offset aO_4 ; "-o"
mov     [ebp+var_40], offset aStratumTcpPool ; "stratum+tcp://pool.supportxmr.com:80"
mov     [ebp+var_3C], offset aU_4 ; "-u"
mov     [ebp+var_38], offset a49wzduvq1dfwg3_4 ; "49WZduVQ1DFWG3scZxFT8hBY1JsoYuJVqMRe8UA"
mov     [ebp+var_34], offset aP_1 ; "-p"
mov     [ebp+var_30], offset asc_51FC28 ; "x"
mov     [ebp+var_2C], offset aO_5 ; "-o"
mov     [ebp+var_28], offset aStratumTcpMine ; "stratum+tcp://mine.xmrpool.net:80"
mov     [ebp+var_24], offset aU_5 ; "-u"
mov     [ebp+var_20], offset a49wzduvq1dfwg3_5 ; "49WZduVQ1DFWG3scZxFT8hBY1JsoYuJVqMRe8UA"
mov     [ebp+var_1C], offset aP_5 ; "-p"
mov     [ebp+var_18], offset asc_51FCBC ; "x"
mov     [ebp+var_14], offset aK ; "-k"
mov     [ebp+var_DC], eax
```

Figure 6 - XMRig config

The configuration is as follows:

1. Monero address (user): 49WZduVQ1DFWG3scZxFT8hBY1JsoYuJVqMRe8UAiYzc2WmGbN7yFDmmc2GZzrAv6GkY24hR7imhNaWME9wEKWPGF3h2FXQB
2. Mining pools:
    a. nanopool.org
    b. supportxmr.com
    c. xmrpool.net

With this Monero user, we performed a search across the aforementioned mining pools, and found the Threat Actors who control it have received, in total, at least 652 XMR (worth $82,176 at this time) in rewards for mining on hijacked systems. The first transaction we can see for this user dates back a few years.

Figure 7 - XMR payout for nanopool.org

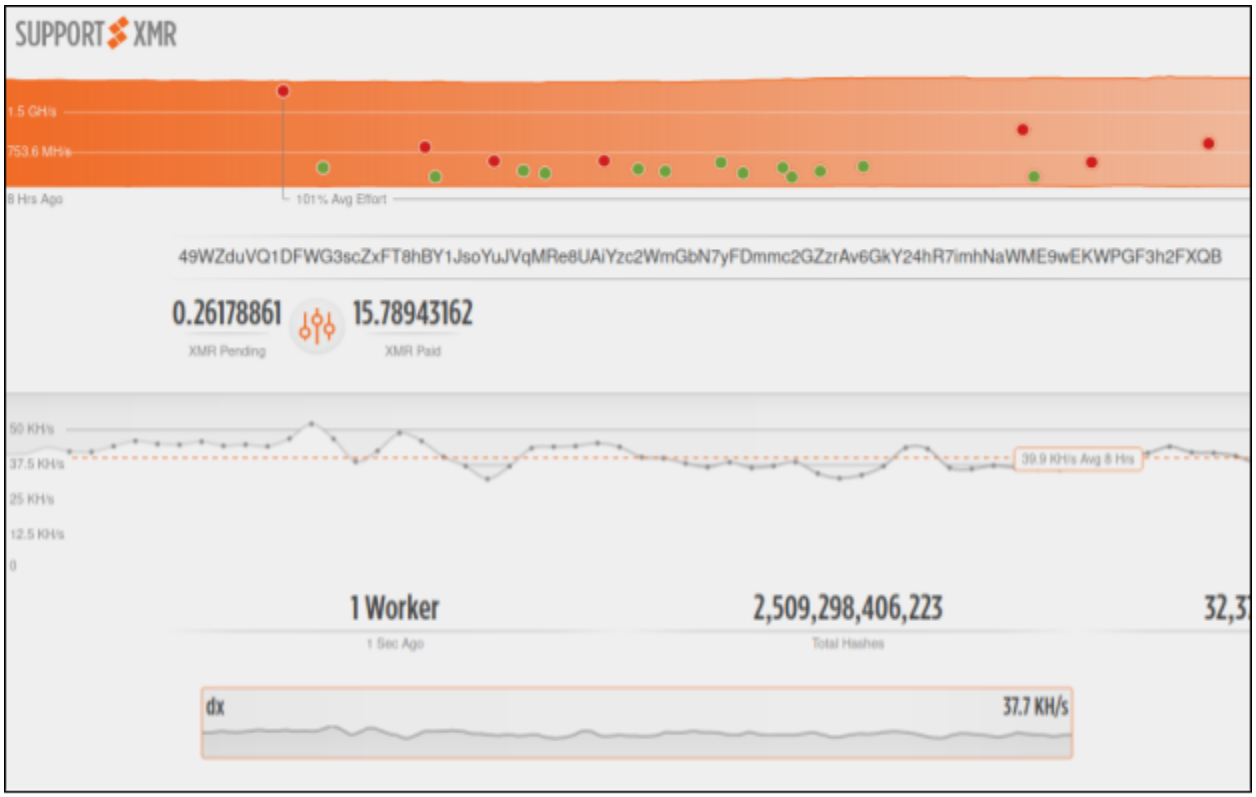

Figure 8 - supportxmr.com payout

Conclusion

In this post we explored the inner-workings of a commonly abused crypto-miner, found installed on a system in which we suspect was exploited through a Confluence Server / Data Center vulnerability. It is our hope that this post informed you of attack techniques and procedures to be on the lookout for.

Observables

Note:
Observables are properties (such as an IP address, MD5 hash, or the value of a registry key) or measurable events (such as the creation of a registry key or a user) and are not indicators of compromise. The observables listed below are intended to provide contextual information only. Deepwatch evaluates the observables and applies those it deems appropriate to our detections.

Observing sets of these properties (observables) could be an indicator of compromise. For instance, observing an IP address, creation of a user with admin privileges and a registry key could be indicators of compromise and should be investigated further.

Observables

Description	Value
Primary mining pool	nanopool.org
Alt mining pool	supportxmr.com
Alt mining pool	xmrpool.net
XMRig unpacked	c95c70b3f884759a968b339787374910ffc8e396b47aafef71ab4f359ee28873
XMRig loader	bdb3c52c9494f5cb79d83fb979c74a08c0c1937e2a949e3bc8d79d5b1994975e