



Earlier this year, [redacted] encountered a relatively new **ransomware** threat actor that called themselves BianLian. We observed the actor deploying custom malware that was written in the Go programming language, which posed some initial, but not insurmountable, reverse-engineering challenges.

BianLian used subtle techniques to exploit, enumerate, and move laterally in victim networks to remain undetected and aggressively worked to counter Endpoint Detection & Response (EDR) protections during the encryption phase of their operations. The group has displayed signs of being new to the practical business aspects of ransomware and associated logistics. Generally they seemed to be experiencing the growing pains of a group of talented hackers new to this aspect of criminal extortion.

Infrastructure associated with the BianLian group first appeared online in December 2021 and their toolset appears to have been under active development since then. Finally, **we have observed the BianLian threat**



The BianLian group has successfully targeted the ProxyShell vulnerability chain (CVE-2021-34473, CVE-2021-34523, CVE-2021-31207) to gain initial access into victim networks. After exploitation, they deployed either a webshell or a lightweight remote access solution such as ngrok as the follow-on payload. BianLian has also targeted SonicWall VPN devices for exploitation, another common target for ransomware groups. Finally, while we do not have direct evidence of a successful attack, we have indications that the actor targets servers that provide remote network access via solutions such as Remote Desktop, attempting to exploit weak or exposed credentials. We have also observed dwell times of up to six weeks from the actor gaining initial access and the actual encryption event.

With a beachhead established within a network, BianLian have shown themselves to be adept with the Living off the Land (LOL) methodology to move laterally, adjusting their operations based on the capabilities and defenses they encountered in the network. For example, they leveraged a combination of the Non-Sucking Service Manager nssm.exe and the reverse proxy ngrok.exe to create backdoors on the servers. Next, they leveraged RDP, WinRM, WMI, and PowerShell to achieve network profiling and lateral movement. Finally, they deployed their custom backdoor to a subset of compromised hosts to provide additional network access should their primary means be disrupted.

As BianLian would initially spread throughout a network, hunting for the most valuable data to steal and identify the most critical machines to encrypt, they appeared to take steps to minimize observable events. As an example, we have observed the **threat actor** choosing to avoid pinging a target and instead utilizing the arp command in network segments where the targeted host would be reachable. In instances where ping was necessary, the actor was judicious in the use, often sending just a single ping. While it is possible that a network defense solution could be configured in such a way to identify an abnormal ping, it is unlikely most common EDR and network security solutions would identify an actor performing targeted network reconnaissance via arp.

Once the BianLian actor identified a host they wished to access, they most often utilized standard LOL techniques such as net.exe to add and/or modify user permissions, netsh.exe to configure host firewall policies, and reg.exe to adjust various registry settings related to remote desktop and security policy enforcement.



```
C:\Windows\system32\net.exe localgroup RemoteDesktopUsers /add  
name to existing admin> /add
```

- "C:\Windows\system32\netsh.exe" advfirewall firewall add rule "name=allow RemoteDesktop" dir=in protocol=TCP localport=3389 action=allow
- "C:\Windows\system32\reg.exe" add
"HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Terminal Server" /v
fAllowToGetHelp /t REG_DWORD /d 1 /f

Even in the final hours prior to encryption, we observed the actor taking care to avoid detection. In one instance, the actor accessed the victim network to seemingly perform last minute network reconnaissance and/or target verification, again sending single pings and arp requests to hosts. The actor then disconnected from the network for approximately an hour before returning to begin their ransom attack in earnest.

Once BianLian made the decision that it was time to encrypt a victims network, they set aside their desire to remain undetected and took a much more aggressive approach, attacking any network and/or host based defense that impeded their custom encryptor tool.

- "C:\Windows\system32\Dism.exe" /online /Disable-Feature
/FeatureName:Windows-Defender /Remove /NoRestart
- [Ref].Assembly.GetType('System.Management.Automation.AmsiUtils').GetField('amsiInitFailed','NonPublic,Static').SetValue(\$null,\$true)
- "C:\Windows\system32\reg.exe" ADD
"HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Sophos Endpoint
Defense\TamperProtection\Config" /t REG_DWORD /v SAVEnabled /d 0 /f



/ C REG_DWORD /v ENABLED /d 0 /f

In one instance, BianLian encountered a server that was configured and defended in such a manner the actor was unable to successfully execute their encryptor. To overcome this, the actor installed TightVNC, modified a registry key to enable network access for TightVNC while in safe mode, then booted the server into safe mode. Since most security applications do not execute in safe mode, this enabled partial encryption of the server.

```
■ "C:\Windows\system32\reg.exe" copy
  hk1m\system\CurrentControlSet\services\tnsvserver
  hk1m\system\CurrentControlSet\control\safeboot\network\tnsvserver /s /f
```

In situations where the actor was able to overcome a victim's defenses, BianLian utilized many of the common techniques observed in a modern ransomware attack such as deleting shadow copy files, deleting backups, as well as distributing and executing their custom encryptor via methods such as RDP, WMI, WinRM, and PowerShell scripts.

In the hour before attempting the encryption phase of an attack, BianLian leveraged LOL tools to prime the network and targeted machines for attack in a less-alerting manner. They created administrator accounts on multiple servers using net.exe and dropped known-good binaries such as 7zip and winscp to enable last-minute data file exfiltration. When the actor started encryption operations, they moved aggressively and with speed. In 30 minutes, [redacted] witnessed dozens of attempts to encrypt a handful of servers, with each attempt blocked by EDR/AV. The actor then spent the next few hours both trying to circumvent security controls and gain access to additional servers that were not initially targeted in an attempt to successfully encrypt the victims files.

Time (Duration)	Event
Start to End (4.5 Hours)	Traffic flowed externally to several different internal user endpoints throughout the entire encryption phase. The majority of malicious traffic to victim servers for the duration of the incident flowed through these same endpoints.



Start+1 Hour (15 Minutes)	admin account passwords changed to hinder any defensive response. Significant account manipulation continued through the event, but was heaviest in these 15 minutes.
Start+1.25 Hours (45 Minutes)	LOL tools for file exfiltration and remote access dropped to multiple servers and executed.
Start+2 Hours (30 Minutes)	Dozens of attempts to encrypt a handful of different servers.
Start+2.5 Hours (2 Hours)	Attempts to circumvent security controls including EDR.
Start+2.5 Hours (2 Hours)	Over a dozen attempts to encrypt several additional servers that were not included in the initial targeting.

The BianLian group has developed a custom tool set consisting of a backdoor and an encryptor, developing both using the Go programming language.

As first highlighted by [MalwareHunterTeam](#), BianLian's custom encryptor was developed in Go. This encryptor also appears to have been under active development since the BianLian group first came online earlier this year. As the MalwareHunterTeam noted, the samples highlighted in VirusTotal contain apparent versioning information:

- `jack/Projects/project1/crypt27`
- `jack/Projects/project1/crypt28`



SHA256: b60be0b5c6e553e483a9ef9040a9314dd54335de7050fed691a07f299ccb8bc6

```
Go buildinf:
go1.18.2
path
project1/crypt8
project1/crypt8
(devel)
project1/common
v0.0.0
../common
(devel)
build
-compiler=gc
build
-gcflags=all=-trimpath=/home/jack/Projects/project1/crypt8
build
-ldflags="-H windowsgui -s -w -extldflags \"-static\""
build
CGO_ENABLED=0
build
GOARCH=amd64
```

As the actor has evolved this encryptor, so has the text used in the ransom note left behind on a victim's computers. While the file name has remained constant, the level of detail and professionalism of the text has improved over time.

```
To unblock your data write us to Tox(https://qtox.github.io/)
████████████████████████████████████████████████████████████████████████████████ or
████████████████████████████████████████████████████████████████████████████████ @mail2tor.com

Don't try to recover yourself otherwise you may lose your data.
Your ID: ████████████████████████████████████████████████████████████████████████████
On the ████████████████████████████████████████████████████████████████████████████ your data will appear on big darknet market if you won't connect us
```



Add user with the following ID to get your instructions:

Alternative way: [redacted]@onionmail.org

Your ID: [redacted]

You should know that we have been downloading data from your network for a significant time before the attack: financial, client, business, post, technical and personal files.

In 10 days – it will be posted at our site [http://bianlian\[redacted\].onion](http://bianlian[redacted].onion) with links send to your clients, partners, competitors and news agencies, that will lead to a negative impact on your company: potential financial, business and reputational loses.

As was reported by [Cyble](#), BianLian's custom encryptor operates on a file extension exclusion model. Once again, we can see that the actor has made adjustments to their binary in that the file types they target as they attempt to ransom victim networks has changed over time.

Compared to Version 27/28, Version 8 of the encryptor would not exclude .lnk files, but would additionally exclude files with the following extensions:

- .drv
- .bianlian
- .mui



```
v1[1] = 4LL;
*v1 = ".exe";
v1[3] = 4LL;
v1[2] = ".sys";
v1[5] = 4LL;
v1[4] = ".drv";
v1[7] = 4LL;
v1[6] = ".dll";
v1[9] = 5LL;
v1[8] = ".html";
v1[11] = 4LL;
v1[10] = ".txt";
v1[13] = 9LL;
v1[12] = ".bianlian";
v1[15] = 4LL;
v1[14] = ".mui";
qword_61C268 = 8LL;
qword_61C270 = 8LL;
if ( dword_6710D0 )
    runtime_gcWriteBarrier();
,
```

In addition to the encryptor, BianLian has developed a simple yet effective backdoor that they have deployed on machines within victim networks, enabling additional means of access. While some actors choose to deploy a full feature remote access tool with a multitude of built-in commands, BianLian's backdoor is, at the core of it, an efficient mechanism for them to retrieve an arbitrary payload from their C2 servers, load it into memory and then execute it.

Each backdoor binary would be configured with a hardcoded IP and Port combination that it will attempt to communicate with. As an example, the binary below will attempt to establish a secure connection to 209.141.54[.]205 on port 5307.

SHA256: da7a959ae7ea237bb6cd913119a35baa43a68e375f892857f6d77eaa62aabbaf

```
/home/admin/prjct/golang/socks/out/209.141.54.205/5307-39718/client/main.go
```

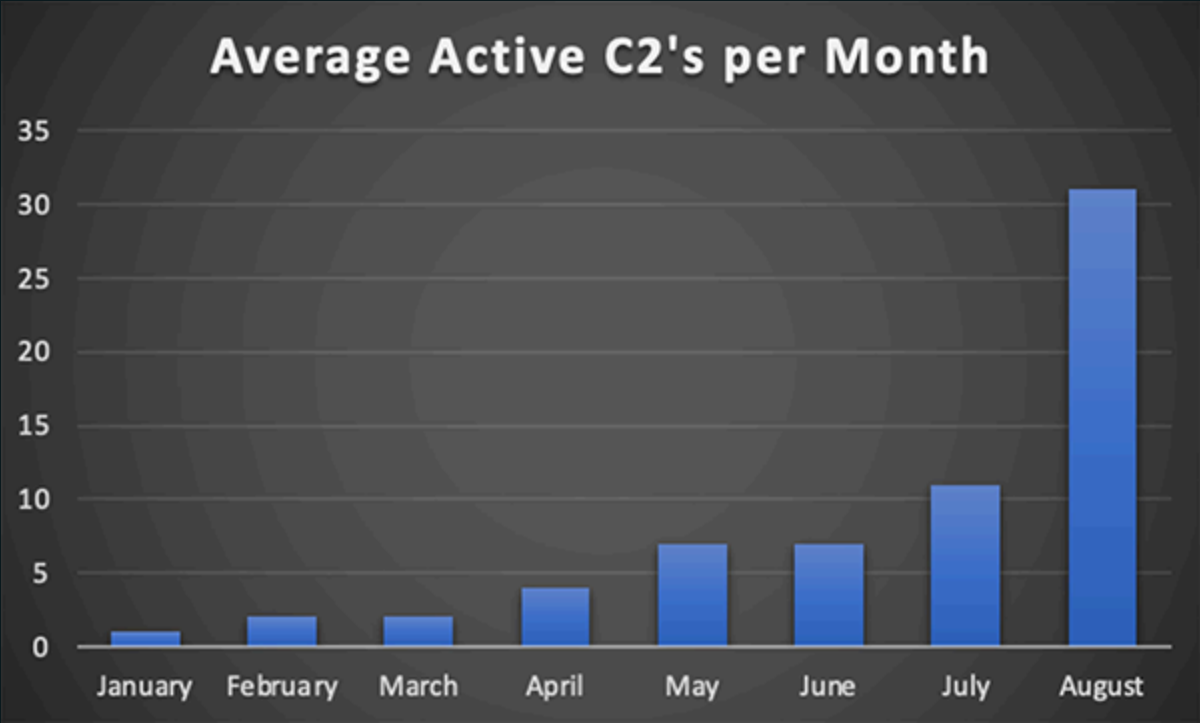
We have observed BianLian deploy multiple backdoors into a victim network with each backdoor configured to either talk to a unique IP or a common IP but on different ports. Not only did this IP and port diversity provide the actor with multiple network paths into the victims network, but every binary would have a unique hash, defeating attempts to detect the backdoors via a simple checksum-based rule.



In investigating BianLian's infrastructure, it appeared the group prefers Linux based hosts for their C2 servers, but we have also found evidence of Windows servers being utilized in their operations. While we do not have enough evidence to confidently identify the C2 software the group is using, we have seen indications that the C2 component is also written in the actors preferred language, Go, which would presumably allow them to easily deploy their C2 solution on either OS.

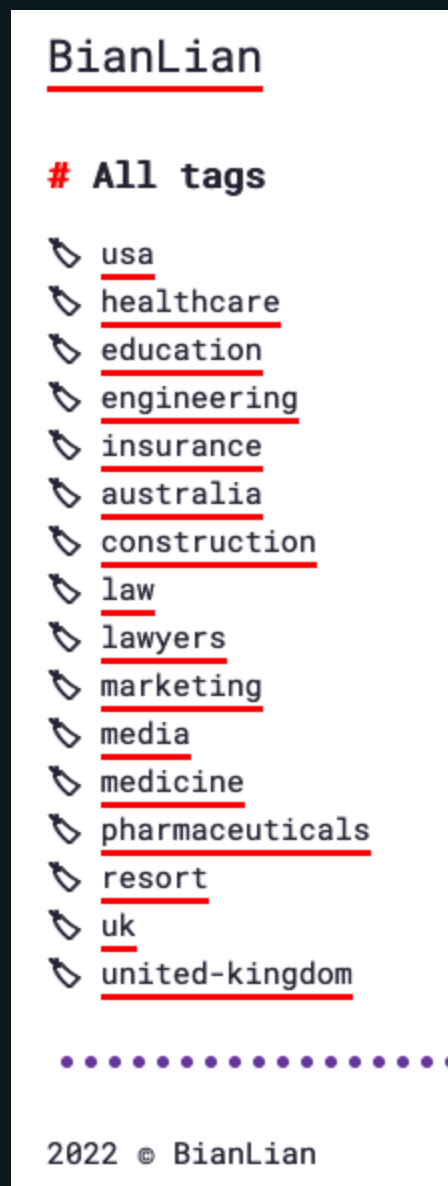
The number of active C2 nodes has also increased in relative relation to the development of the actor's toolkit. Based on our research, the earliest known C2 server we have identified, 23.94.56[.]154, first appeared online at the end of December 2021 and remained active until early August of this year. From that initial IP, BianLian appeared to have gradually acquired new C2 servers, occasionally removing an IP, before reaching approximately ten active servers by the end of July.

Starting in August, we observed what appeared to be a somewhat troubling explosion in the rate by which BianLian was bringing new C2 servers online. Throughout the month, BianLian continued to add new C2 nodes to their operational infrastructure, ending the month with approximately 30 active IPs, a three-fold increase in just a matter of weeks. While we lack the insight to know the exact cause for this sudden explosion in growth, this may signal that they are ready to increase their operational tempo, though whatever the reason, there is little good that comes from a ransomware operator having more resources available to them.





As is the norm for a group conducting double extortion style ransomware attacks, the BianLian group maintained a leak site where they post the data they have exfiltrated from victim networks. While an unfortunate truth in the ransomware space is that the true number of organizations and victims of ransomware attacks will never be known, as of September 1, 2022, the BianLian site has posted details on twenty victim organizations. The threat actor also took the time to categorize the industry vertical of the victims and tagged the corresponding data.



The victim organizations range from small/medium size businesses to a large multinational company, with the majority of the companies based in North America, the UK and Australia.



attempt to have them pay the actors ransom demand.

We also note that this is a small sample size and continued observation will be required before drawing significant conclusions on victimology or any possible preference in targeting by BianLian.

While there is a long history of seemingly new ransomware groups rising from the ashes of defunct and/or rebranded groups, we do not have any indications at this time to suggest that is the case with BianLian. For all intents and purposes, the BianLian group appears to represent a new entity in the ransomware ecosystem. Furthermore, we assess that the BianLian actors represent a group of individuals who are very skilled in network penetration but are relatively new to the extortion/ransomware business. This hypothesis is based in part on our observations of how the BianLian group has managed the business side of a ransomware operation compared to their relative skill level in compromising and navigating victim networks.

While the actor has proven themselves proficient at compromising a victim network, we have seen the actor:

- Mistakenly sending data from one victim to another.
- Possessing a relatively stable backdoor toolkit, but have an actively developing encryption tool with an evolving ransom note.
- Long delays in communications with victims.
- Through the groups own admission on their onion site, the business side of their infrastructure is unreliable.

BianLian

[Home](#) [Companies](#) [Tags](#) [Contacts](#)

Currently we're updating our infrastructure. So you may have temporary issues to download huge files.

Note: There is an Android banking trojan that has been referred to by some researchers as BianLian (a.k.a. Hydra.) To date, we have seen no indications that this is related to the BianLian ransomware group. [1](#) [2](#)



techniques, but also preparing to act quickly and effectively when a compromise inevitably happens.

This includes ensuring you have:

- An aggressive, prioritized patching regime;
- Employ multi-factor authentication on every system that allows that as an option;
- Visibility into your network and endpoint devices to quickly identify breaches;
- Secure backups to allow return to business operations as soon as possible;
- A well practiced incident response plan so everyone involved knows their role; and
- An assessment of your 'Crown Jewels' that can be used to both inform your security posture and decide ahead of an incident what data you could afford to have leaked so you can avoid paying the ransom.

In addition to these strategic recommendations, there are multiple opportunities for behavioral detections in the attack chain leveraged by BianLian:

1. Defense Evasion: Svchost not a child of services.exe

- BianLian called one of their LOL tools svchost, then launched it via a process other than services.exe.

2. Defense Evasion: Svchost executing from an unusual path

- BianLian called one of their LOL tools svchost.exe, then executed it from a non-standard path.

3. Defense Evasion: Netsh to modify firewall rules

- BianLian leveraged netsh to add a firewall rule to open 3389 to Remote Desktop.

4. Reconnaissance: Ping -4 -n 1

- BianLian used single pings to perform network reconnaissance. This is a false-positive prone alert.

5. Lateral Movement: Winrm dropping a file via PowerShell

- The binary wsmprovhost.exe is used to mediate the relationship between WinRM and PowerShell. Alerting on file modification by wsmprovhost.exe proved a reliable method to detect



- If leveraging an EDR that classifies binaries as known and unknown and ties network connections to binaries, looking for 3389 in use by unknown binaries can be extremely fruitful. This rule detects BianLian’s custom Go backdoor.
7. Credential Access: Account manipulation via net.exe
- “Net user” is too loud to alert on in most environments, but we recommend alerting on a threshold of “net user” executions. Even a threshold as high as 10 events in 15 minutes would have detected BianLian in the attacks witnessed.
8. Execution: Unknown binary launching PowerShell
- If leveraging an EDR that classifies binaries as known and unknown, searching for unknown binaries launching PowerShell will frequently detect use of the BianLian backdoor
9. Defense Evasion: Reg.exe modifying safeboot keys
- BianLian added a remote access tool to safeboot keys in order to enable network access for their remote access tool in safeboot.

- 001f33dd5ec923afa836bb9e8049958decc152eeb6f6012b1cb635cff03be2a2
- 1a1177363be7319e7fb50ac84f69acb633fd51c58f7d2d73a1d5efb5c376f256
- 20bab94e6d9c8ed4832ce3b58f9150b16f9e5f40ffdc747e10366cab5a30352
- 36281d02e28dd26a1db37ebe36941fc9eb1748868e96b544f227b3b59de51fea
- 3bdcc81931687abac9e6ba4c80d4d596cebb470c80f56213aa29d3da43925537
- 50c86fb27bed1962903a5f9d155544e3fdb859ae19e967a10f0bf3a60bb8954f
- 5d429e05cede806ecea2e99116cac09558fcc0011095201e66c2e65c42f80fcf
- 64065c29b369881ee36314c0d15e442510027186fd9087aec0f63e22a5c6f24c
- 6d7009df2fa033f7adc30793ebd5254ef47a803950e31f5c52fa3ead1197599f
- 8084eddfdb157edf8b1c0cdf8bf4d4e4aaa332fc871c2892aa4113b5148ac63e
- 8592862cd28bcc23cfbcf57c82569c0b74a70cd7ea70dbdee7421f3fafc7ecaf
- 86a9b84c6258c99b3c3c5b94a2087bc76a533f6043829ded5d8559e88b97fb2f
- 9b7a0117a27dc418fbf851afcd96c25c7ad995d7be7f3d8d888fa26a6e530221



cd17afd9115b2d83e948a1bcabf508f42d0fe7edb56cc62f5cc467c938e45033
d602562ba7273695df9248a8590b510ccd49fefb97f5c75d485895abba13418d
da7a959ae7ea237bb6cd913119a35baa43a68e375f892857f6d77eaa62aabbaf
dda89e9e6c70ff814c65e1748a27b42517690acb12c65c3bbd60ae3ab41e7aca
de31a4125eb74d0b7cbf2451b40fdb2d66d279a8b8fd42191660b196a9ac468f
f7a3a8734c004682201b8873691d684985329be3fcdba965f268103a086ebaad

1fd07b8d1728e416f897bef4f1471126f9b18ef108eb952f4b75050da22e8e43
b60be0b5c6e553e483a9ef9040a9314dd54335de7050fed691a07f299ccb8bc6
cbab4614a2cdd65eb619a4dd0b5e726f0a94483212945f110694098194f77095
eaf5e26c5e73f3db82cd07ea45e4d244ccb3ec3397ab5263a1a74add7bbcb6e2

104.225.129[.]86
104.238.223[.]10
104.238.223[.]3
109.248.6[.]207
13.49.57[.]110
144.208.127[.]119
146.0.79[.]9
157.245.80[.]66
16.162.137[.]220
165.22.87[.]199
172.93.96[.]61
172.93.96[.]62
18.130.242[.]71
185.108.129[.]242
185.225.69[.]173
185.56.80[.]28
185.62.58[.]151
185.69.53[.]38
192.145.38[.]242



■ 45.9.150[.]132

■ 5.2.79[.]138

■ 51.68.190[.]20

■ 54.173.59[.]51

■ 62.84.112[.]68

■ 64.52.80[.]120

■ 66.135.0[.]42

■ 83.136.180[.]12

■ 85.13.117[.]213

■ 85.13.117[.]218

■ 91.199.209[.]20

■ 95.179.137[.]20

■ 104.207.155[.]133

■ 104.238.61[.]153

■ 146.70.44[.]248

■ 155.94.160[.]241

■ 167.88.15[.]98

■ 172.96.137[.]107

■ 188.166.81[.]141

■ 194.26.29[.]131

■ 194.5.212[.]205

■ 194.58.119.159

■ 198.252.108[.]34

■ 202.66.72[.]7

■ 208.123.119[.]145

■ 209.141.54[.]205

■ 23.227.198[.]243

■ 23.94.56[.]154

■ 43.155.116[.]250

■ 45.144.30[.]139

■ 45.92.156[.]105



- 85.13.117[.]219
- 89.22.224[.]3

IP Address	First Seen
104.225.129[.]86	Late July
104.238.223[.]10	Late July
104.238.223[.]3	Late August
109.248.6[.]207	Late August
13.49.57[.]110	Mid May
144.208.127[.]119	Mid August
146.0.79[.]9	Early February
157.245.80[.]66	Early June
16.162.137[.]220	Mid July
165.22.87[.]199	Late August
172.93.96[.]61	Mid August



18.130.242[.]71	Mid July
185.108.129[.]242	Early August
185.225.69[.]173	Mid August
185.56.80[.]28	Early August
185.62.58[.]151	Mid August
185.69.53[.]38	Late May
192.145.38[.]242	Late August
192.161.48[.]43	Mid August
192.169.6[.]232	Mid August
37.235.54[.]81	Late August
45.9.150[.]132	Early August
5.2.79[.]138	Late August
51.68.190[.]20	Late August
54.173.59[.]51	Mid August
62.84.112[.]68	Mid August



66.135.0[.]42	Early April
83.136.180[.]12	Early June
85.13.117[.]213	Late August
85.13.117[.]218	Late August
91.199.209[.]20	Mid July
95.179.137[.]20	Late July

IP Address	First Seen	Last Seen
104.207.155[.]133	Early July	Early August
104.238.61[.]153	*	
146.70.44[.]248	*	
155.94.160[.]241	Late July	Early August
167.88.15[.]98	Early August	Late August
172.96.137[.]107	Early August	Mid August
188.166.81[.]141	Early May	Late August



194.5.212[.]205	Early August	Mid August
194.58.119[.]159	Late May	Early June
198.252.108[.]34	*	
202.66.72[.]7	Mid August	Late August
208.123.119[.]145	Late April	Late April
209.141.54[.]205	Early August	Mid August
23.227.198[.]243	*	
23.94.56[.]154	Late December	Early August
43.155.116[.]250	Mid August	Mid August
45.144.30[.]139	Mid April	Early June
45.92.156[.]105	*	
5.188.6[.]118	Early August	Late August
5.230.67[.]2	Early August	Late August
85.13.116[.]194	Mid August	Late August
85.13.117[.]219	Early August	Late August



* These IPs were found in instances of BianLian's backdoor, but we lack visibility on the timeframe(s) when the IPs may have been active.

- `"C:\Windows\system32\Dism.exe" /online /Disable-Feature /FeatureName:Windows-Defender /Remove /NoRestart`
- `"C:\Windows\system32\net.exe" localgroup "Remote Desktop Users" <similar name to existing admin> /add`
- `"C:\Windows\system32\net.exe" user <legitimate admin account> 3gDZNxtsQ9G029k7D6Ljxe /domain`
- `"C:\Windows\system32\netsh.exe" advfirewall firewall set rule "group=remote desktop" new enable=Yes`
- `"C:\Windows\system32\netsh.exe" advfirewall firewall add rule "name=allow RemoteDesktop" dir=in * protocol=TCP localport=3389 action=allow`
- `"C:\Windows\system32\reg.exe" add "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Terminal Server" /* v fAllowToGetHelp /t REG_DWORD /d 1 /f`
- `"C:\Windows\system32\reg.exe" add "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Terminal Server\WinStations\RDP-Tcp" /v UserAuthentication /t REG_DWORD /d 0 /f`
- `"C:\Windows\system32\reg.exe" ADD "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Sophos Endpoint * Defense\TamperProtection\Config" /t REG_DWORD /v SAVEnabled /d 0 /f`
- `"C:\Windows\system32\reg.exe" ADD "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Sophos Endpoint * Defense\TamperProtection\Config" /t REG_DWORD /v SEDEnabled /d 0 /f`
- `"C:\Windows\system32\reg.exe" ADD * HKEY_LOCAL_MACHINE\SOFTWARE\WOW6432Node\Sophos\SAVService\TamperProtection /t REG_DWORD /v Enabled /d 0 /* f`



```
HKLM\SYSTEM\CurrentControlSet\Control\SafeBoot\Network\CVIServer /S /I

■ \cmd.exe /Q /c net user "Administrator" /active:yes 1>
  \\127.0.0.1\C$\Windows\Temp\abjAlC 2>&1

■ cmd.exe /Q /c net user "Administrator" ChangeMe2morrow! 1>
  \\127.0.0.1\C$\Windows\Temp\OxNEcz 2>&1

■ cmd.exe /Q /c quser 1> \\127.0.0.1\C$\Windows\Temp\VXPrvY 2>&1

■ "C:\Windows\system32\PING.EXE" -4 -n 1 *

■ [Ref].Assembly.GetType('System.Management.Automation.AmsiUtils').GetField('
  amsiInitFailed','NonPublic,* Static').SetValue($null,$true)
```

ID	Technique
T1190	Initial Access: Exploit Public-Facing Application
T1047	Execution: Windows Management Instrumentation
T1059.001	Execution: Command and Scripting Interpreter: PowerShell
T1098	Persistence: Account Manipulation
T1078	Persistence: Valid Accounts
T1562.001	Defense Evasion: Impair Defenses: Disable or Modify Tools



T1036	Defense Evasion: Masquerading
T1112	Defense Evasion: Modify Registry
T1069	Discovery: Permission Groups Discovery
T1018	Discovery: Remote System Discovery
T1021.001	Lateral Movement: Remote Services: Remote Desktop Protocol
T1021.005	Lateral Movement: Remote Services: VNC
T1021.006	Lateral Movement: Remote Services: Windows Remote Management
T1090	Command and Control: Proxy
T1071.001	Command and Control: Application Layer Protocol: Web Protocol
T1486	Impact: Data Encrypted for Impact

During our research, we created some tool modifications for the [AlphaGolang](#) project to assist other security community researchers in working on the BianLian malware. The specific update is located [here](#).

Threat Intelligence



Ben Armstrong is a Senior Threat Intelligence Researcher at [redacted] specializing in both cybercrime and nation-state threat actors. Prior to [redacted], Ben was on the Cyber Intelligence Team at Walmart.



Lauren serves as the Director of Incident Response and Forensics at [redacted] where she's frequently found on the front lines, leading incident response efforts on behalf of clients. Prior to joining [redacted], Lauren worked at Los Alamos National Laboratory where she specialized in malware analysis as a member and occasional leader of the incident response team. She enjoys



in Computer Engineering, Computer Science and a PhD in International Affairs, an Honorary State University.



Twitter

Senior Threat Intelligence Analyst at [redacted] who spends his days trying to find the badness lurking in the tubes.



Twitter

Danny Quist is the Director of Special Projects at [redacted]. He works on the research team developing new methods of reverse engineering, machine learning, and malware detonation. Previously he has worked for MIT Lincoln Laboratory and Los Alamos National Laboratory. He has presented at Blackhat, RSA, Defcon, and DFRWS.



—

—



—



© Copyright Redacted, Inc. 2023

[Privacy](#)