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

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.

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.

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.

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

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

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`

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.

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-A2F3-9B070C86E261}" /TR "cm
```

```
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] [DOMAIN NAME] Admini
```

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 :

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:

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.

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.

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.

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

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

```
C:\redacted\find.exe  -f objectcategory=computer -csv name cn OperatingSyste
```

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

```
C:\Windows\system32\cmd.exe /C wmic /namespace:\\root\SecurityCenter2 PATH A
C:\Windows\system32\cmd.exe /C wmic /namespace:\\root\SecurityCenter2 PATH A
C:\Windows\system32\cmd.exe /C wmic /namespace:\\root\SecurityCenter2 PATH F
```

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 powershell -nop -w hidden -
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 Event ID:7045
```

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

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

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\Terminal Server\WinStations\
```

Makes sure the RDP listener is enabled.

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

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

```
REG  ADD "HKLM\SYSTEM\CurrentControlSet\Control\Terminal Server" /t REG_DWOR
```

Makes sure the terminal server is set to enabled.

```
REG ADD "HKLM\SYSTEM\CurrentControlSet\Control\Terminal Server" /t REG_DWORD
```

Makes sure terminal services is set to remote admin mode.

```
REG ADD "HKLM\SYSTEM\CurrentControlSet\Control\Terminal Server" /t REG_DWORD
```

Makes sure that the terminal service will start idle sessions.

```
REG ADD "HKLM\SYSTEM\CurrentControlSet\Control\Terminal Server" /t REG_DWORD
```

Enables advertisement of the terminal server.

```
REG ADD "HKLM\SYSTEM\CurrentControlSet\Control\Terminal Server" /t REG_DWORD
```

Makes sure terminal server is set to allow connections.

```
REG ADD "HKLM\SYSTEM\CurrentControlSet\Control\Terminal Server" /t REG_DWORD
```

Makes sure terminal server is set to simultaneous sessions.

```
REG ADD HKLM\SYSTEM\CurrentControlSet\Control\Terminal Server\Licensing Core
```

Makes sure multiple sessions are allowed.

```
REG ADD "HKLM\SYSTEM\CurrentControlSet\Control\Terminal Server" /t REG_DWORD
```

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
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-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:f2:0a:00:ad:ac:ce
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:b3:72:01:29:74:c4:87:db:15:d7
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:ff:bf:47:aa:04:5c ]
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[.]com,ns2.dxabt[.]com,ns3
Public Algorithm: rsaEncryption
JA3: 0eecb7b1551fba4ec03851810d31743f
JA3s: ae4edc6faf64d08308082ad26be60767
```

Config:

```
{
    "x64": {
        "uri_queried": "/tRPG",
        "sha256": "dec25fc2fe7e76fe191fbfdf48588c4325f52bfe2769fbc88a5614541
        "config": {
            "HTTP Method Path 2": "/faq",
            "Jitter": 79,
            "C2 Server": "dxabt[.]com,/case",
            "Spawn To x86": "%windir%\\syswow64\\runonce.exe",
            "Method 1": "GET",
            "C2 Host Header": "",
            "Method 2": "POST",
            "Watermark": 426352781,
            "Spawn To x64": "%windir%\\sysnative\\runonce.exe",
            "Beacon Type": "8 (HTTPS)",
            "Port": 443,
            "Polling": 53988
        },
        "time": 1637416040175.3,
        "md5": "30cc71d5b5d7778774c54486558690d3",
        "sha1": "5f36c6cffdbae0d631c8889b4d9bad1248f899b3"
    },
    "x86": {
        "uri_queried": "/Mr0m",
        "sha256": "a992d57b2f6164e599952ea3c245962824ad17166684ed45e987efe80
        "config": {
            "HTTP Method Path 2": "/faq",
            "Jitter": 79,
            "C2 Server": "dxabt[.]com,/case",
            "Spawn To x86": "%windir%\\syswow64\\runonce.exe",
            "Method 1": "GET",
            "C2 Host Header": "",
            "Method 2": "POST",
            "Watermark": 426352781,
            "Spawn To x64": "%windir%\\sysnative\\runonce.exe",
            "Beacon Type": "8 (HTTPS)",
            "Port": 443,
            "Polling": 53988
        },
        "time": 1637416038974.9,
        "md5": "c1fd49c043894c1dff8bc02b17f8942c",
        "sha1": "e915f74be310b1687db6b290af2f78583a981512"
    }
}
```

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.

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:e2bc969424adc97345ac81194d316f58da38621aad3ca7ae27e40a8fae582987
```

```
QBot DLL (extracted from registry):
MD5:312e52b4109741893f17bc524084100f
SHA1:7ca650945223eab088f43fd472e3592be2ed9d32
SHA2:4d3b10b338912e7e1cbade226a1e344b2b4aebc1aa2297ce495e27b2b0b5c92b

cool.exe
MD5:59E7F22D2C290336826700F05531BD30
SHA1:3B2A0D2CB8993764A042E8E6A89CBBF8A29D47D1
SHA256:F63E17FF2D3CFE75CF3BB9CF644A2A00E50AAFFE45C1ADF2DE02D5BD0AE35B0
```

Detections

Network

```
ET POLICY Powershell Activity Over SMB - Likely Lateral Movement
ET POLICY Command Shell Activity Using Comspec Environmental Variable Over S
ET RPC DCERPC SVCCTL - Remote Service Control Manager Access
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 are disabled by defaul
ET EXPLOIT Possible Zerologon NetrServerReqChallenge with 0x00 Client Challe
ET EXPLOIT Possible Zerologon NetrServerAuthenticate with 0x00 Client Creden
ET EXPLOIT [401TRG] Possible Zerologon (CVE-2020-1472) UUID flowbit set - 20
ET EXPLOIT [401TRG] Possible Zerologon (CVE-2020-1472) M2 - 2030889
```

New signatures thanks to @ET_Labs!

```
2035258 - ET EXPLOIT Zerologon Phase 2/3 - NetrServerAuthenticate2 Request w
2035259 - ET EXPLOIT Zerologon Phase 2/3 - NetrServerAuthenticate2 Request w
2035260 - ET EXPLOIT Zerologon Phase 2/3 - NetrServerAuthenticate3 Request w
2035261 - ET EXPLOIT Zerologon Phase 2/3 - NetrServerAuthenticate3 Request w
```

```
2035262 - ET EXPLOIT Zerologon Phase 3/3 - Malicious NetrServerPasswordSet2
2035263 - ET EXPLOIT Zerologon Phase 3/3 - NetrLogonSamLogonWithFlags Reques
```

Sigma

```
title: Scheduled task executing powershell encoded payload from registry
status: Experimental
description: Detects the creation of a schtask that executes a base64 encode
author: @Kostastsale, @TheDFIRReport
references:
  - https://thedfirreport.com/2022/02/21/qbot-and-zerologon-lead-to-full-dom
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 ZeroLogon PoC execut
author: @Kostastsale, @TheDFIRReport
references:
  - https://thedfirreport.com/2021/11/01/from-zero-to-domain-admin/
  - https://thedfirreport.com/2022/02/21/qbot-and-zerologon-lead-to-full-dom
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 execution
status: Experimental
description: Detects the execution of reg.exe and subsequent command line ar
author: @Kostastsale, @TheDFIRReport
references:
  - https://thedfirreport.com/2022/02/21/qbot-and-zerologon-lead-to-full-dom
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 Server'
      - '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 Winstations2) or (selection2
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-and-zerologon-lead-t
*/


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


import "pe"


rule qbot_8734_payload_dll {
   meta:
      description = "files - file e2bc969424adc97345ac81194d316f58da38621aad
      author = "The DFIR Report"
      reference = "https://thedfirreport.com"
      date = "2022-02-20"
      hash1 = "e2bc969424adc97345ac81194d316f58da38621aad3ca7ae27e40a8fae582
   strings:
      $s1 = "Terfrtghygine.dll" fullword ascii
      $s2 = "Winamp can read extended metadata for titles. Choose when this 
      $s3 = "Read metadata when file(s) are loaded into Winamp" fullword wid
      $s4 = "Use advanced title formatting when possible" fullword wide /* G
      $s5 = "PQVW=!?" fullword ascii
      $s6 = "Show underscores in titles as spaces" fullword wide /* Goodware
      $s7 = "Advanced title display format :" fullword wide /* Goodware Stri
      $s8 = "CreatePaint" fullword ascii
      $s9 = "PQRVW=2\"" fullword ascii
      $s10 = "Advanced Title Formatting" fullword wide /* Goodware String - 
      $s11 = "Read metadata when file(s) are played or viewed in the playlis
      $s12 = "Show '%20's in titles as spaces" fullword wide /* Goodware Str
      $s13 = "Example : \"%artist% - %title%\"" fullword wide /* Goodware St
      $s14 = "PQRVW=g" fullword ascii
      $s15 = "PQRW=e!" fullword ascii
      $s16 = "ATF Help" fullword wide /* Goodware String - occured 1 times *
      $s17 = "(this can be slow if a large number of files are added at once
      $s18 = "PQRVW=$" fullword ascii
      $s19 = "Metadata Reading" fullword wide /* Goodware String - occured 1
      $s20 = "Other field names: %artist%, %album%, %title%, %track%, %year%
   condition:
      uint16(0) == 0x5a4d and filesize < 2000KB and
      ( pe.imphash() == "aa8a9db10fba890f8ef9edac427eab82" and pe.exports("C
}


rule qbot_dll_8734 {
   meta:
      description = "files - qbot.dll"
      author = "TheDFIRReport"
      reference = "QBOT_DLL"
      date = "2021-12-04"
      hash1 = "4d3b10b338912e7e1cbade226a1e344b2b4aebc1aa2297ce495e27b2b0b5c
   strings:
      $s1 = "Execute not supported: %sfField '%s' is not the correct type of
      $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 ascii
      $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 encoded string '|' */
```

```
       $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

Share this:

Twitter LinkedIn Reddit Facebook WhatsApp

« QBOT LIKES TO MOVE IT, MOVE IT

2021 YEAR IN REVIEW »

Search …

Search

Type your email…

Subscribe

Register For Our Next CTF

Reports

Threat Intelligence

Detection Rules

DFIR Labs

Mentoring and Coaching