

Qbot

Qbot Likes to Move It, Move It

February 7, 2022

[Qbot](#) (aka QakBot, Quakbot, Pinkslipbot) has been around for a long time having first been observed back in 2007. More info on Qbot can be found at the following links: [Microsoft](#) & [Red Canary](#).

In this case, from October 2021, we will break down how Qbot quickly spread across all workstations in an environment, while stealing browser information and emails. While the case is nearly 5 months old, Qbot infections in the past week have followed the same pattern.

Case Summary

We did not observe the initial access for this case but assess with medium to high confidence that a malicious email campaign was used to deliver an Excel (xls) document. Following the opening of the xls document, the initial Qbot DLL loader was downloaded and saved to disk. Interestingly, the name of the DLL contained a .html extension to disguise the portable executable nature of the payload. Once executed, the Qbot process creates a scheduled task to elevate itself to system.

Qbot injected into many processes but one favorite in this intrusion, was Microsoft Remote Assistance (msra.exe). Within minutes of landing on the beachhead, a series of discovery commands were executed using Microsoft utilities. Around the same time, LSASS was access by Qbot to collect credentials from memory.

Thirty minutes after initial access, Qbot was observed collecting data from the beachhead host including browser data and emails from Outlook. At around 50 minutes into the infection, the beachhead host copied a Qbot dll to an adjacent workstation, which was then executed by remotely creating a service. Minutes later, the beachhead host did the same thing to another adjacent workstation and then another, and before we knew it, all workstations in the environment were compromised.

Qbot followed it’s normal process on each machine. Servers were not accessed in this intrusion. After this activity, normal beaconing occurred but no further actions on objectives were seen.

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, Shocker, 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.

Search

Search

Sélectionner une langue ▾

Fourni par Google Traduction

Subscribe



Register For
Our Next
CTF



Reports

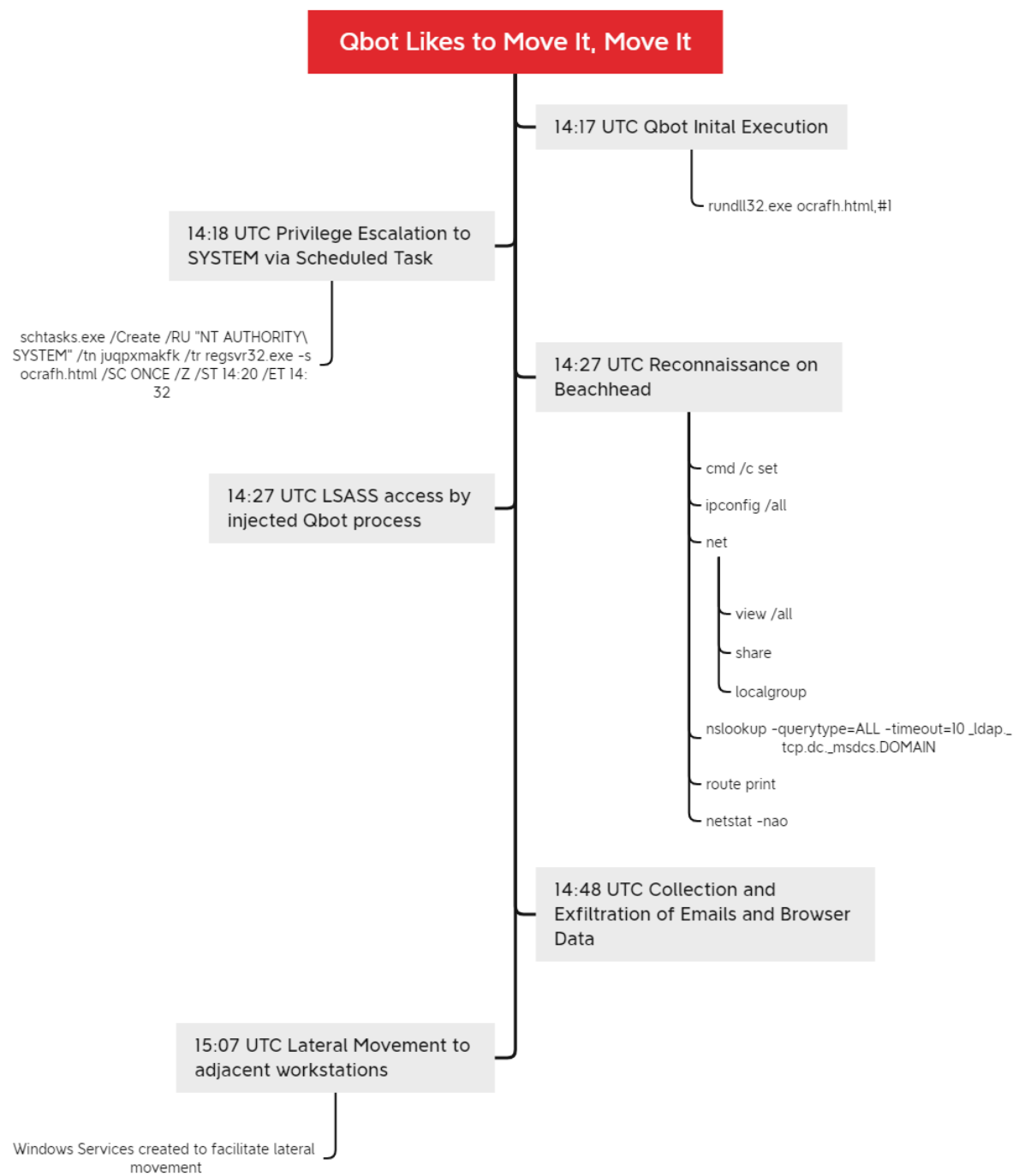


Threat
Intelligence



Detection
Rules

Timeline



Analysis and reporting completed by [@iiamaleks](#)

Reviewed by and [@MetallicHack](#) & [@tas_kmanager](#)

MITRE ATT&CK

Initial Access

We assess with medium to high confidence that the QBot infection was delivered to the system via a malspam campaign through a hidden 4.0 Macro’s in Excel.

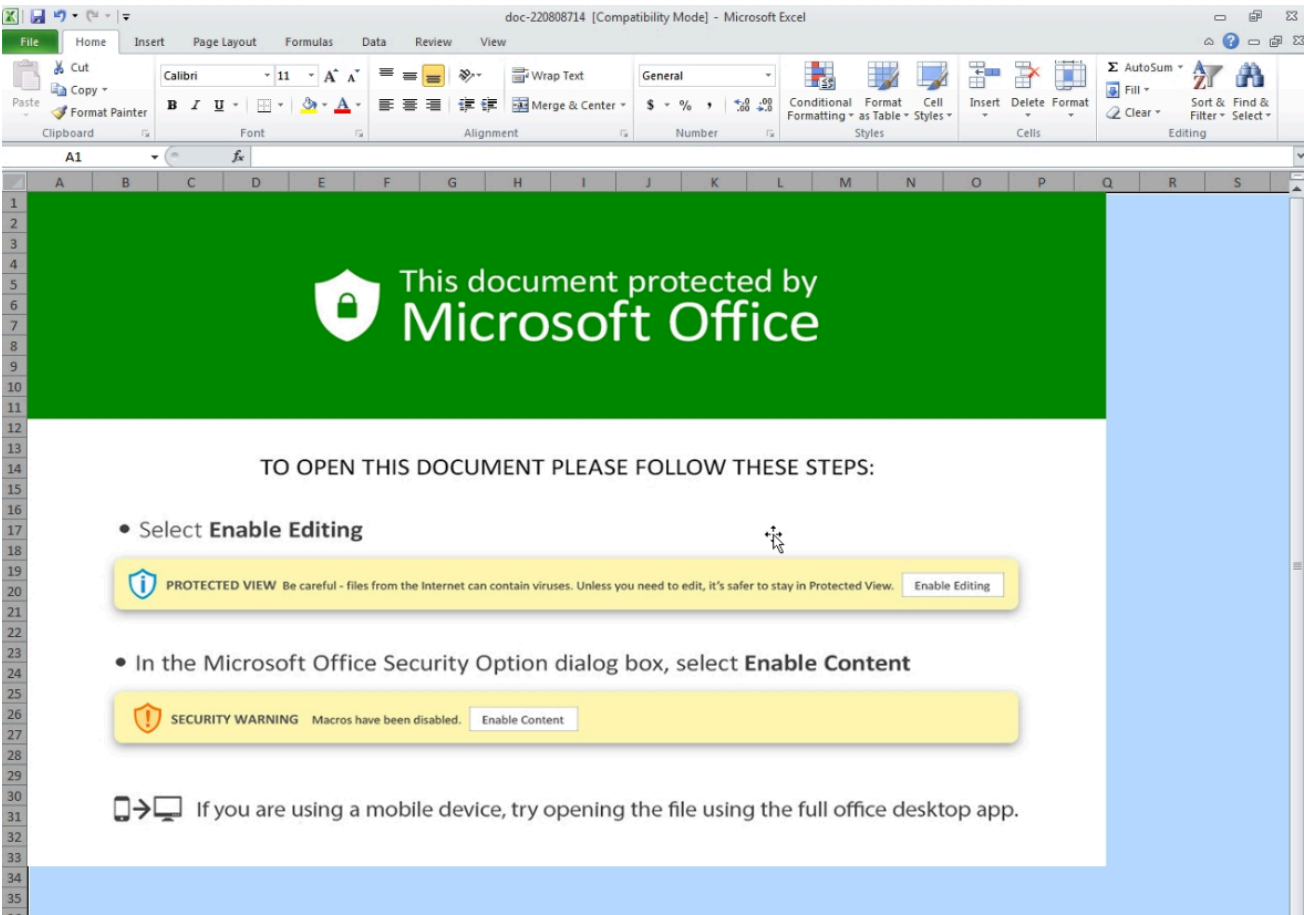
We believe [this](#) is the xls file that lead to the Qbot infection, due to the overlap in time period, download url, and file name.



DFIR Labs



Mentoring and Coaching



Execution

The QBot dll was executed on the system and shortly after, injected into the msra.exe process.

Action Type	Initiating Process Command Line	Process Command Line
ProcessCreated	"rundll32.exe" C:\Users\ [REDACTED] \ocrafh.html,#1	"rundll32.exe" C:\Users\ [REDACTED] \ocrafh.html,#1
NtMapViewOfSectionRemoteApiCall	"rundll32.exe" C:\Users\ [REDACTED] \ocrafh.html,#1	msra.exe

Privilege Escalation

A scheduled task was created by Qbot to escalate to SYSTEM privileges. This scheduled task was created by the msra.exe process, to be run only once, a few minutes after its creation.

Action Type	Initiating Process Command Line
ProcessCreated	"schtasks.exe" /Create /RU "NT AUTHORITY\SYSTEM" /tn juqpxmakfk /tr "regsvr32.exe -s \"%C:\Users\ [REDACTED] \ocrafh.html\"\" /SC ONCE /Z /ST 14:20 /ET 14:32

```
"schtasks.exe" /Create /RU "NT AUTHORITY\SYSTEM" /tn juqpxmakfk /tr "r
```

Defense Evasion

QBot was observed injecting into msra.exe process on multiple systems.

Action Type	Initiating Process Command Line	Process Command Line
ProcessCreated	"rundll32.exe" C:\Users\ [REDACTED] \Downloads\ocrafh.html,#1	msra.exe
NtMapViewOfSectionRemoteApiCall	"rundll32.exe" C:\Users\ [REDACTED] \Downloads\ocrafh.html,#1	msra.exe

Multiple folders were added to the Windows Defender Exclusions list in order to prevent the Qbot dll placed inside of it from being detected. The newly dropped dll was then executed and process injected into msra.exe.

Qbot used reg.exe to add Defender folder exceptions for folders within AppData and ProgramData.

```
C:\Windows\system32\reg.exe ADD \\"HKLM\SOFTWARE\Microsoft\Microsoft Antimalware
```

```
C:\Windows\system32\reg.exe ADD \"HKLM\SOFTWARE\Microsoft\Windows Defender\Exc.
```

dll files dropped by Qbot, were deleted after injection into msra.exe.

Credential Access

LSASS was accessed by Qbot, with the intention of accessing credentials. This can be observed through the Sysmon process access event, indicating the GrantedAccess value of 0x1410.

Additional evidence of LSASS access was visible in API calls from Qbot injected processes to LSASS.

Discovery

The following discovery commands where observed coming from the Qbot processes. These commands where executed on the beachhead system along with other workstations compromised through lateral movement.

```
whoami /all
arp -a
cmd /c set
arp -a
net view /all
ipconfig /all
net view /all
nslookup -querytype=ALL -timeout=10 _ldap._tcp.dc._msdcs.REDACTED
route print
net share
net1 localgroup
net localgroup
netstat -nao
```

Lateral Movement

Qbot moved laterally to all workstations in the environment by copying a dll to the machine and then remotely creating a service to execute the Qbot dll. The services created had the DeleteFlag set causing the service to be removed upon reboot.

The following occurred on each workstation:

The lateral movement activity from the beachhead host was rapid and connections were seen across all workstations in the network. A view from the memory of the beachhead host shows the injected msra process connecting to hosts across the network.

The service creations were also observed via event id 7045 across all hosts.

Collection

Qbot is widely known to [steal emails](#) with the intention of collecting information and performing email thread hijacking.

Email data will be collected and stored in 1 of 2 locations.

```
C:\Users\Username\EmailStorage_ComputerHostname-Username_TimeStamp
C:\Windows\system32\config\systemprofile\EmailStorage_ComputerHostname-Username
```

Once exfiltrated from the system this folder is then deleted as seen below

```
cmd.exe /c rmdir /S /Q "C:\Users\REDACTED\EmailStorage_REDACTED-REDACTED"
cmd.exe /c rmdir /S /Q "C:\Windows\system32\config\systemprofile\EmailStorage"
```

Collection of browser data from Internet Explorer and Microsoft Edge was also observed with Qbot using the built-in utility esentutl.exe.

```
esentutl.exe /r V01 /l"C:\Users\REDACTED\AppData\Local\Microsoft\Windows\Internet Explorer\History\HistoryIndex.dat"
```

Command and Control

Qbot uses a tiered infrastructure, often using other compromised systems as first tier proxy points for establishing a constantly changing list of C2 endpoints. You can review a in-depth analysis of the modules of this malware in this [Checkpoint report](#).

With this type of setup the list of C2 from October 2021, has in large rotated out of use. To keep up to date on current Qbot C2 endpoints you can check out our [Threat Feed & All Intel service](#) as we track these changing lists daily.

Qbot does use SSL in it's C2 communication but does not rely solely on port 443 for communication, in the case investigated here the following ports were found in the extracted [C2 configuration](#).

Count	Port
88	443
25	995
17	2222
3	2078
2	465
2	20
1	993
1	61201
1	50010
1	32100
1	21
1	1194

Qbot uses SSL and while the domains do not resolve, they follow a pattern and are detectable with several Suricata ETPRO signatures.

Qbot JA3/S:

```
JA3: 72a589da586844d7f0818ce684948eea, c35a61411ee5bdf666b4d64b05c29e64
JA3s: 7c02dbae662670040c7af9bd15fb7e2f
```

Impact

The final actions of the threat actor were not observed, however, the data exfiltrated from the network could be used to conduct further attacks or sold to 3rd parties.

IOCs

Network

120.150.218.241:995
71.74.12.34:443
24.229.150.54:995

185.250.148.74:443
136.232.34.70:443
82.77.137.101:995
75.188.35.168:443
72.252.201.69:443
109.12.111.14:443
68.204.7.158:443
196.218.227.241:995
27.223.92.142:995
76.25.142.196:443
73.151.236.31:443
185.250.148.74:2222
173.21.10.71:2222
189.210.115.207:443
105.198.236.99:443
47.22.148.6:443
24.55.112.61:443
24.139.72.117:443
45.46.53.140:2222
92.59.35.196:2222
95.77.223.148:443
68.186.192.69:443
89.101.97.139:443
173.25.166.81:443
140.82.49.12:443

File

ocrafh.html.dll
2897721785645ad5b2a8fb524ed650c0
d836fa75f0682b4c393418231aefca97169d551e
956ecb4afa437eafe56f958b34b6a78303ad626baee004715dc6634b7546bf85
qbbwlwjmlmnaggd.dll
e0fafe1b4eb787444ed457dbf05895a4
16b5b1494e211b74e97d9f35ff5a994f70411f2e
9f6e3b0b18f994950b40076d1386b4da4ce0f1f973b129b32b363aac4a678631
hyietnrfrx.uit
b6ed9b2819915c2b57d4c58e37c08ba4
e9ff9b7e144bdad9d8955f4a328f7b6daa2b455e
70a49561f39bb362a2ef79db15e326812912c17d6e6eb38ef40343a95409a19a
znmxbx.evj
2a8cf6154e6a129ffd07a501bbc0b098
304d8e812a8d988e21af8a865d8dd577dc6f3134
e510566244a899d6a427c1648e680a2310c170a5f25aff53b15d8de52ca11767
zsokarzi.xpq
43660d21bfa1431e0ee3426cd12ddf38
5d3b7e0c05e65aa0dfc8b5e48142d782352e36be
cbfc135bff84d63c4a0ccb5102cfa17d8c9bf297079f3b2f1371dafcbefea77c
tuawktso.vbe
ad413cd422c1a0355163618683e936a0
5fca07dfc68a13b3707636440d5c416e56149357
1411250eb56c55e274fbcf0741bbd3b5c917167d153779c7d8041ab2627ef95f
jtrbde.dll
5dd964c8d9025224eb658f96034babea
6c526a28ed49b2ef83548e20a71610877e69d450
3d913a4ba5c4f7810ec6b418d7a07b6207b60e740dde8aed3e2df9ddf1caab27
rzmulxiilw.dll
000df43b256cdc27bb22870919bb1dfa

f94d5bf14dee6a6e8db957d49c259082dd82350b
ca564c6702d5e653ed8421349f4d37795d944793a3dbd1bb3c5dbc5732f1b798
ljncxcwmsg.gjf
88834d17d2cdce884a73e38638a4e0dd
b5b264d00a7d6d6b3dd4965dbe2bd00e0823ba6c
c789bb45cacf0de1720e707f9edd73b4ed0edc958b3ce2d8f0ad5d4a7596923a

Detections

Network

ETPRO TROJAN Observed Qbot Style SSL Certificate
ETPRO TROJAN Possible Qbot SSL Cert
ET POLICY PE EXE or DLL Windows file download HTTP

Sigma

```
title: QBot process creation from scheduled task REGSVR32 (regsvr32.exe)
id: 33d9c3f4-57a6-4ddb-a2a0-b2ccf8482607
status: test
description: Detects the process creation from Scheduled Task with REGSVR32
author: tas_kmanager, TheDFIRReport
references:https://thedfirreport.com/2022/02/07/qbot-likes-to-move-it-move-it-
date: 2022/02/06
modified: 2022/02/06
logsource:
category: process_creation
product: windows
detection:
selection:
CommandLine|contains|all:
- 'schtasks.exe'
- 'regsvr32.exe -s'
- 'SYSTEM'
condition: selection
falsepositives:
- unknown
level: high
tags:
- attack.persistence
- attack.privilege_escalation
- attack.t1053.005
- qbot
```

```
title: QBot scheduled task REGSVR32 and C$ image path
id: 014da553-5727-4e47-9544-56da83b3eb6f
description: Detects the creation of Scheduled Task with REGSVR32 (regsvr32.exe)
status: test
author: tas_kmanager, TheDFIRReport
references:https://thedfirreport.com/2022/02/07/qbot-likes-to-move-it-move-it-
date: 2022/02/06
modified: 2022/02/06
logsource:
product: windows
service: system
detection:
selection:
```

```
Provider_Name: 'Service Control Manager'
EventID: 7045
ImagePath|contains|all:
- 'regsvr32.exe'
- 'C$'
condition: selection
level: high
falsepositives:
- low
tags:
- attack.persistence
- attack.privilege_escalation
- attack.t1053.005
- qbot
```

```
title: EmailStorage file deletion - QBot
id: 695e7200-c733-44b3-9231-6d3459c668ba
status: test
description: Detect EmailStorage file deletion after QBot infection
author: tas_kmanager, TheDFIRReport
references:https://thedfirreport.com/2022/02/07/qbot-likes-to-move-it-move-it-
date: 2022/02/06
modified: 2022/02/06
logsource:
category: process_creation
product: windows
detection:
selection:
ParentCommandLine|contains:
- '\EmailStorage_'
- 'rmdir'
Image|endswith: '\cmd.exe'
condition: selection
falsepositives:
- low
level: high
tags:
- attack.defense_evasion
- attack.t1070.004
- qbot
```

[Whoami Execution Anomaly](#)
[Suspicious Reconnaissance Activity](#)
[Mimikatz Detection LSASS Access](#)

Yara

```
/*
YARA Rule Set
Author: The DFIR Report
Date: 2022-02-07
Identifier: Case 7685
Reference: https://thedfirreport.com/2022/02/07/qbot-likes-to-move-it-move-it-
*/

/* Rule Set ----->
```

```
import "pe"

rule tuawktso_7685 {
  meta:
    description = "Files - file tuawktso.vbe"
    author = "The DFIR Report"
    reference = "https://thedfirreport.com"
    date = "2022-02-01"
    hash1 = "1411250eb56c55e274fbcf0741bbd3b5c917167d153779c7d8041ab2627ef951"
  strings:
    $s1 = "* mP_5z" fullword ascii
    $s2 = "44:HD:\\C" fullword ascii
    $s3 = "zoT.tid" fullword ascii
    $s4 = "dwmcoM<" fullword ascii
    $s5 = "1iHBuSER:" fullword ascii
    $s6 = "78NLog.j" fullword ascii
    $s7 = "-FtP4p" fullword ascii
    $s8 = "x<d%[ * " fullword ascii
    $s9 = "02f+ " fullword ascii
    $s10 = "- wir2" fullword ascii
    $s11 = "+ \"z?}xn$" fullword ascii
    $s12 = "+ $Vigb" fullword ascii
    $s13 = "# W}7k" fullword ascii
    $s14 = "# N)M)9" fullword ascii
    $s15 = "?uE- d0" fullword ascii
    $s16 = "W_* 32" fullword ascii
    $s17 = ">v9+ H" fullword ascii
    $s18 = "tUg$* h" fullword ascii
    $s19 = "`\"*- M" fullword ascii
    $s20 = "b^D$ -L" fullword ascii
  condition:
    uint16(0) == 0xe0ee and filesize < 12000KB and
    8 of them
}

rule wmyvpa_7685 {
  meta:
    description = "Files - file wmyvpa.sae"
    author = "The DFIR Report"
    reference = "https://thedfirreport.com"
    date = "2022-02-01"
    hash1 = "3d913a4ba5c4f7810ec6b418d7a07b6207b60e740dde8aed3e2df9ddf1caab2"
  strings:
    $s1 = "spfX.hRN<" fullword ascii
    $s2 = "wJriR>E0ODA[.tIM" fullword ascii
    $s3 = "5v:\\VAL" fullword ascii
    $s4 = "K6U:\\&" fullword ascii
    $s5 = "%v,.IlZ\\" fullword ascii
    $s6 = "\\kX>%n -" fullword ascii
    $s7 = "!D1lqj" fullword ascii
    $s8 = "&ZvM* " fullword ascii
    $s9 = "AU8]+ " fullword ascii
    $s10 = "- vt>h" fullword ascii
    $s11 = "+ u4hRI" fullword ascii
    $s12 = "ToX- P" fullword ascii
    $s13 = "S!G+ u" fullword ascii
    $s14 = "y 9-* " fullword ascii
    $s15 = "nl}* J" fullword ascii
```

```
$s16 = "t /Y Fo" fullword ascii
$s17 = "0^w- F" fullword ascii
$s18 = "N -Vw'" fullword ascii
$s19 = "hVHjzI4" fullword ascii
$s20 = "ujrej8" fullword ascii
condition:
    uint16(0) == 0xd3c2 and filesize < 12000KB and
    8 of them
}

rule ocrafh_html_7685 {
    meta:
        description = "Files - file ocrafh.html.dll"
        author = "The DFIR Report"
        reference = "https://thedfirreport.com"
        date = "2022-02-01"
        hash1 = "956ecb4afa437eafe56f958b34b6a78303ad626baee004715dc6634b7546bf85"
    strings:
        $s1 = "Over.dll" fullword wide
        $s2 = "c:\\339\\Soon_Back\\Hope\\Wing\\Subject-sentence\\Over.pdb" fullword
        $s3 = "7766333344" ascii /* hex encoded string 'wf33D' */
        $s4 = "6655557744" ascii /* hex encoded string 'fUwD' */
        $s5 = "7733225566" ascii /* hex encoded string 'w3"Uf' */
        $s6 = "5577445500" ascii /* hex encoded string 'UwDU' */
        $s7 = "113333" ascii /* reversed goodwill string '333311' */
        $s8 = "'56666" fullword ascii /* reversed goodwill string '66665'' */
        $s9 = "224444" ascii /* reversed goodwill string '444422' */
        $s10 = "0044--" fullword ascii /* reversed goodwill string '--4400' */
        $s11 = "444455" ascii /* reversed goodwill string '554444' */
        $s12 = "5555//" fullword ascii /* reversed goodwill string '//5555' */
        $s13 = "44...." fullword ascii /* reversed goodwill string '....44' */
        $s14 = ",,,2255//5566" fullword ascii /* hex encoded string '"Uuf' */
        $s15 = "44//446644//" fullword ascii /* hex encoded string 'DDfD' */
        $s16 = "7755//44----." fullword ascii /* hex encoded string 'wUD' */
        $s17 = "?^.4444--,,55" fullword ascii /* hex encoded string 'DDU' */
        $s18 = "66,,5566////55" fullword ascii /* hex encoded string 'fUfU' */
        $s19 = "operator co_await" fullword ascii
        $s20 = "?\"55////////77" fullword ascii /* hex encoded string 'Uw' */
    condition:
        uint16(0) == 0x5a4d and filesize < 2000KB and
        ( pe.imphash() == "fadf54554241c990b4607d042e11e465" and ( pe.exports("D
    }

rule ljncxcwmsg_7685 {
    meta:
        description = "Files - file ljncxcwmsg.gjf"
        author = "The DFIR Report"
        reference = "https://thedfirreport.com"
        date = "2022-02-01"
        hash1 = "c789bb45cacf0de1720e707f9edd73b4ed0edc958b3ce2d8f0ad5d4a7596923a
    strings:
        $s1 = "x=M:\"*" fullword ascii
        $s2 = "=DdlLxu" fullword ascii
        $s3 = "#+- 7 " fullword ascii
        $s4 = "1CTxH* " fullword ascii
        $s5 = "OF0+ K" fullword ascii
        $s6 = "\\oNvd4Ww" fullword ascii
        $s7 = "jvKSZ21" fullword ascii
        $s8 = "o%U%uhuc]" fullword ascii
```

```

    $s9 = "~rCcqlf1 0" fullword ascii
    $s10 = "kjoYf^=8" fullword ascii
    $s11 = "jpOMR4}" fullword ascii
    $s12 = "ZIIUn'u" fullword ascii
    $s13 = "7uCyy7=H" fullword ascii
    $s14 = "#c.sel}W" fullword ascii
    $s15 = ")t)uSKv%&}" fullword ascii
    $s16 = "VGiAP/o(" fullword ascii
    $s17 = "SwcF~i`" fullword ascii
    $s18 = "*ITDe5\\n" fullword ascii
    $s19 = "MjKB!X" fullword ascii
    $s20 = "tjfvUus" fullword ascii
condition:
    uint16(0) == 0xa5a4 and filesize < 2000KB and
    8 of them
}

rule hyietnrfrx_7685 {
    meta:
        description = "Files - file hyietnrfrx.uit"
        author = "The DFIR Report"
        reference = "https://thedfirreport.com"
        date = "2022-02-01"
        hash1 = "70a49561f39bb362a2ef79db15e326812912c17d6e6eb38ef40343a95409a19a"
    strings:
        $s1 = "Z)* -^'" fullword ascii
        $s2 = "%EGMf%mzT" fullword ascii
        $s3 = "CYR:\"n" fullword ascii
        $s4 = "CbIN$P;" fullword ascii
        $s5 = "We:\\>K" fullword ascii
        $s6 = "h^nd* " fullword ascii
        $s7 = "+ GR;q" fullword ascii
        $s8 = "u%P%r2A" fullword ascii
        $s9 = "ti+ gj?" fullword ascii
        $s10 = "glMNdH8" fullword ascii
        $s11 = "SuiMFrn7" fullword ascii
        $s12 = "K* B5T" fullword ascii
        $s13 = "eLpsNt " fullword ascii
        $s14 = "aQeG% SMF " fullword ascii
        $s15 = "JdYQ67 " fullword ascii
        $s16 = "f>xYrBDvNF+Q" fullword ascii
        $s17 = "OESW[>0" fullword ascii
        $s18 = "9rlPY5__" fullword ascii
        $s19 = "DMvH{ }L" fullword ascii
        $s20 = ".dgQ>H" fullword ascii
    condition:
        uint16(0) == 0x4eee and filesize < 2000KB and
        8 of them
}

rule zsokarzi_7685 {
    meta:
        description = "Files - file zsokarzi.xpq"
        author = "The DFIR Report"
        reference = "https://thedfirreport.com"
        date = "2022-02-01"
        hash1 = "cbfc135bff84d63c4a0ccb5102cfa17d8c9bf297079f3b2f1371dafcbefea77c"
    strings:
        $s1 = "}poSpY" fullword ascii
```

```
$s2 = "[cmD>S" fullword ascii
$s3 = "# {y|4" fullword ascii
$s4 = "IX%k%5u" fullword ascii
$s5 = "YKeial7" fullword ascii
$s6 = "#%y% !" fullword ascii
$s7 = "wOUV591" fullword ascii
$s8 = "| VJHt}&Y" fullword ascii
$s9 = "BEgs% 5" fullword ascii
$s10 = "UKCy\\n" fullword ascii
$s11 = "w;g0xQ?" fullword ascii
$s12 = "'OHSf\"/x" fullword ascii
$s13 = "#qVNkOnj" fullword ascii
$s14 = "{_0qzbVbN" fullword ascii
$s15 = "QEQro\\4" fullword ascii
$s16 = "ohFq\\P" fullword ascii
$s17 = "34eYZVnp2" fullword ascii
$s18 = "rxuqLDG" fullword ascii
$s19 = "kUZI6J#" fullword ascii
$s20 = "IEJl1}+" fullword ascii
condition:
    uint16(0) == 0xc1d7 and filesize < 2000KB and
    8 of them
}


rule znmxbx_7685 {
    meta:
        description = "Files - file znmxbx.evj"
        author = "The DFIR Report"
        reference = "https://thedfirreport.com"
        date = "2022-02-01"
        hash1 = "e510566244a899d6a427c1648e680a2310c170a5f25aff53b15d8de52ca11767"
    strings:
        $s1 = "# /rL,;" fullword ascii
        $s2 = "* m?#;rE" fullword ascii
        $s3 = ">\\'{'6|B{" fullword ascii /* hex encoded string 'k' */
        $s4 = "36\\$'48`" fullword ascii /* hex encoded string '6H' */
        $s5 = "&#$2\\&6&[" fullword ascii /* hex encoded string '&' */
        $s6 = "zduwzpa" fullword ascii
        $s7 = "CFwH}&.MWi " fullword ascii
        $s8 = "e72.bCZ<" fullword ascii
        $s9 = "*c:\"HK!\\\" fullword ascii
        $s10 = "mBf:\"t~" fullword ascii
        $s11 = "7{R:\"O`" fullword ascii
        $s12 = "7SS.koK#" fullword ascii
        $s13 = "7lS od:\\\" fullword ascii
        $s14 = "kMRWSyi$%D^b" fullword ascii
        $s15 = "Wkz=c:\\\" fullword ascii
        $s16 = "1*1:\\\"L" fullword ascii
        $s17 = "GF8$d:\\\"T" fullword ascii
        $s18 = "i$\\".N8spy" fullword ascii
        $s19 = "f4LOg@" fullword ascii
        $s20 = "XiRcwU" fullword ascii
    condition:
        uint16(0) == 0x3888 and filesize < 12000KB and
        8 of them
}
```


MITRE

- Rundll32 – T1218.011
- Scheduled Task – T1053.005
- Disable or Modify Tools – T1562.001
- Process Injection – T1055
- LSASS Memory – T1003.001
- Network Share Discovery – T1135
- Local Groups – T1069.001
- Local Account – T1087.001
- System Network Connections Discovery – T1049
- System Network Configuration Discovery – T1016
- Internet Connection Discovery – T1016.001
- Email Collection – T1114
- Credentials from Web Browsers – T1555.003
- Commonly Used Port – T1043
- Application Layer Protocol – T1071
- Web Protocols – T1071.001
- Exfiltration Over C2 Channel – T1041

Internal case #7685

Share this:

 Twitter

 LinkedIn

 Reddit

 Facebook

 WhatsApp

Related

- Follina Exploit Leads to Domain Compromise
- Qbot and Zerologon Lead To Full Domain Compromise
- 2022 Year in Review

« COBALT STRIKE, A DEFENDER’S GUIDE – PART 2

QBOT AND ZEROLOGON LEAD TO FULL DOMAIN COMPROMISE »