SANS ISC: InfoSec Handlers Diary Blog - SANS Internet Storm Center SANS Site Network Current Site SANS Internet Storm Center Other SANS Sites Help Graduate Degree Programs Security Training Security Certification Security Awareness Training Penetration Testing Industrial Control Systems Cyber Defense Foundations DFIR Software Security Government OnSite Training InfoSec Handlers Diary Blog

isc.sans.edu/diary/rss/26862

Recent Qakbot (Qbot) activity

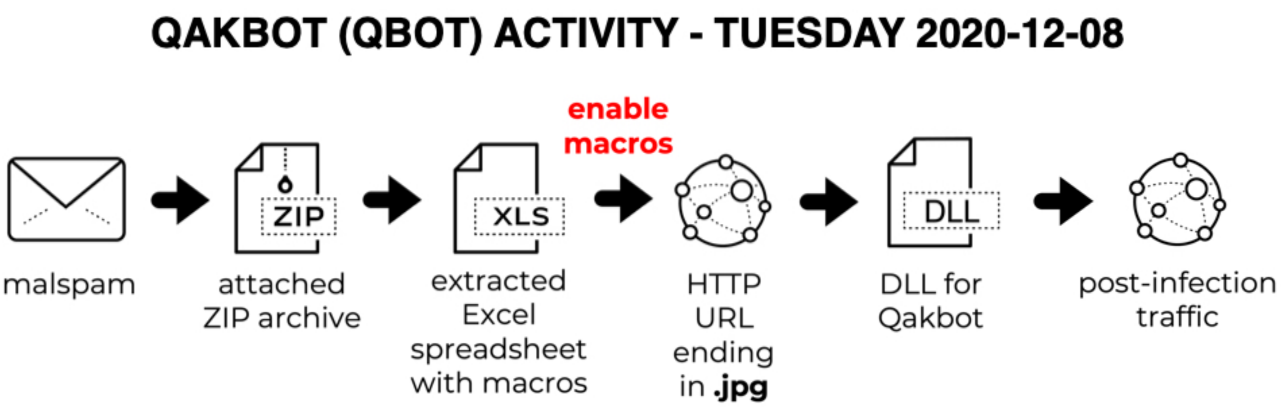
Published: 2020-12-09
Last Updated: 2020-12-09 04:51:14 UTC
by Brad Duncan (Version: 1)
0 comment(s)
Introduction

Today's diary is a review of a Qakbot (Qbot) infection I generated on Tuesday 2020-12-08.

Qakbot generally includes follow-up malware like Cobalt Strike (such as this example), but my infection on Tuesday 2020-12-08 was a basic Qakbot infection that didn't run long enough for follow-up malware or other activity.

Of note, in late-November 2020, Qakbot underwent a version update. I've noticed this in my day-to-day research, but nothing comprehensive has been published yet. A few tweets about it:

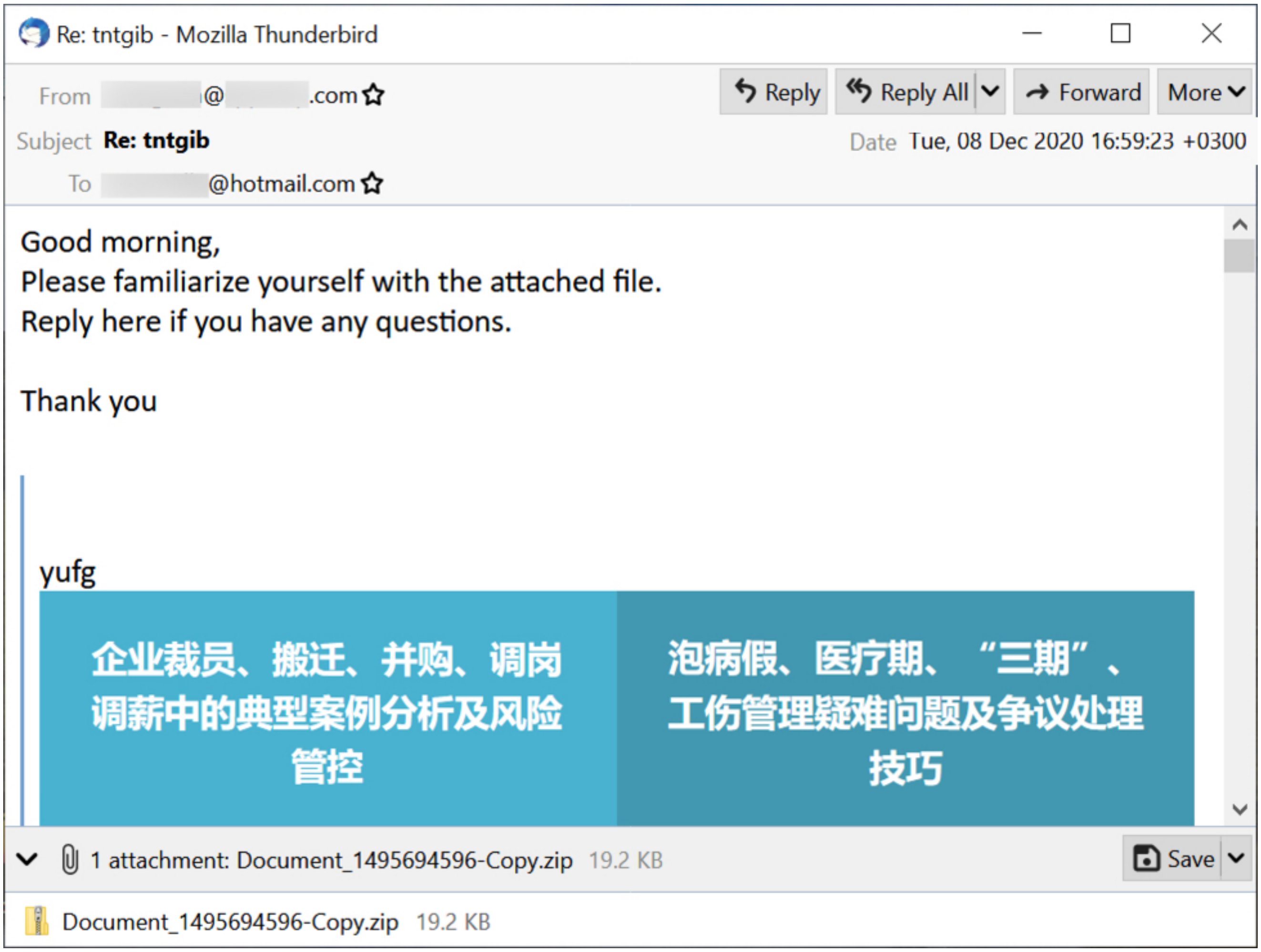
I'll review some of the changes I've noticed about the update in today's diary.



Shown above: Chain of events for the Qakbot infection we're reviewing today.

The malspam

Malspam examples I found from Tuesday 2020-12-08 were fake replies to legitimate email chains, although the example shown below might be a Qakbot-generated reply for an unsolicited spam message.



From [redacted]@[redacted].com

Subject **Re: tntgib**

Date Tue, 08 Dec 2020 16:59:23 +0300

To [redacted]@hotmail.com

Good morning,
Please familiarize yourself with the attached file.
Reply here if you have any questions.

Thank you

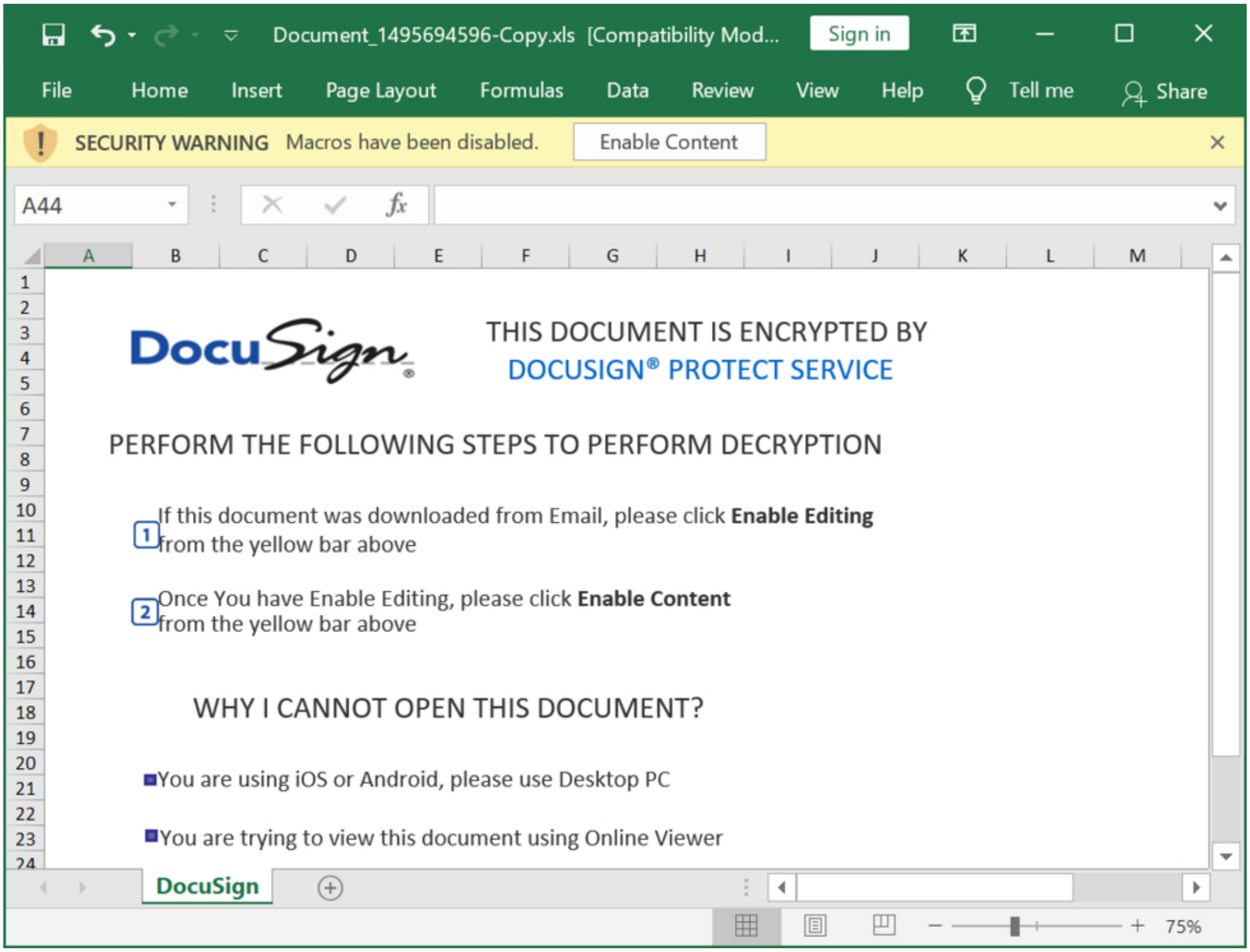
> yufg
>
> 企业裁员、搬迁、并购、调岗调薪中的典型案例分析及风险管控
>
> 泡病假、医疗期、“三期”、工伤管理疑难问题及争议处理技巧

1 attachment: Document_1495694596-Copy.zip 19.2 KB

Document_1495694596-Copy.zip 19.2 KB

Shown above: An example of Qakbot malspam from Tuesday 2020-12-08.

The attached ZIP archive has an Excel spreadsheet with macros designed to infect a vulnerable Windows host with Qakbot malware. Even with the version update, these spreadsheets distributing Qakbot have the same template we've seen for the past several months.



SECURITY WARNING Macros have been disabled.

THIS DOCUMENT IS ENCRYPTED BY
DOCUSIGN® PROTECT SERVICE

PERFORM THE FOLLOWING STEPS TO PERFORM DECRYPTION

1 If this document was downloaded from Email, please click **Enable Editing** from the yellow bar above

2 Once You have Enable Editing, please click **Enable Content** from the yellow bar above

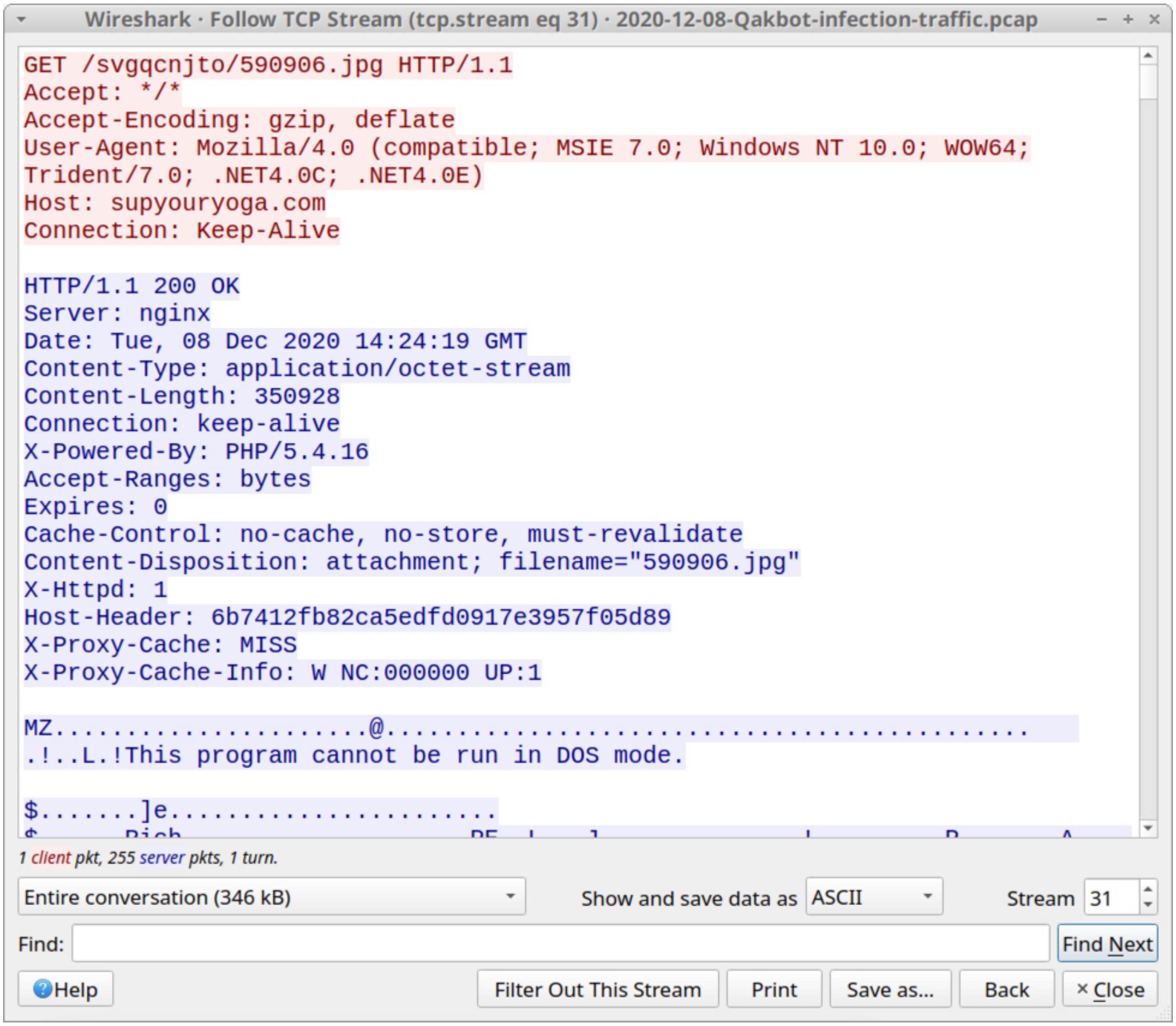
WHY I CANNOT OPEN THIS DOCUMENT?

- You are using iOS or Android, please use Desktop PC
- You are trying to view this document using Online Viewer

Shown above: Excel spreadsheet extracted from the ZIP attachment.

Infection activity

Typical for Qakbot, we see an HTTP GET request for a URL ending in **.jpg** that returned a Windows binary (in this case a DLL). This often is an HTTPS URL, where we would not see the Windows binary in a pcap. In recent moths, I've seen as many HTTPS URLs for this as I have regular HTTP URLs.



Wireshark · Follow TCP Stream (tcp.stream eq 31) · 2020-12-08-Qakbot-infection-traffic.pcap

```
GET /svgqcnjto/590906.jpg HTTP/1.1
Accept: */*
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 10.0; WOW64;
Trident/7.0; .NET4.0C; .NET4.0E)
Host: supyouryoga.com
Connection: Keep-Alive

HTTP/1.1 200 OK
Server: nginx
Date: Tue, 08 Dec 2020 14:24:19 GMT
Content-Type: application/octet-stream
Content-Length: 350928
Connection: keep-alive
X-Powered-By: PHP/5.4.16
Accept-Ranges: bytes
Expires: 0
Cache-Control: no-cache, no-store, must-revalidate
Content-Disposition: attachment; filename="590906.jpg"
X-Httpd: 1
Host-Header: 6b7412fb82ca5edfd0917e3957f05d89
X-Proxy-Cache: MISS
X-Proxy-Cache-Info: W NC:000000 UP:1

MZ......................@..........................................
.!..L.!This program cannot be run in DOS mode.

$.......]e.......................
```

1 client pkt, 255 server pkts, 1 turn.

Entire conversation (346 kB)

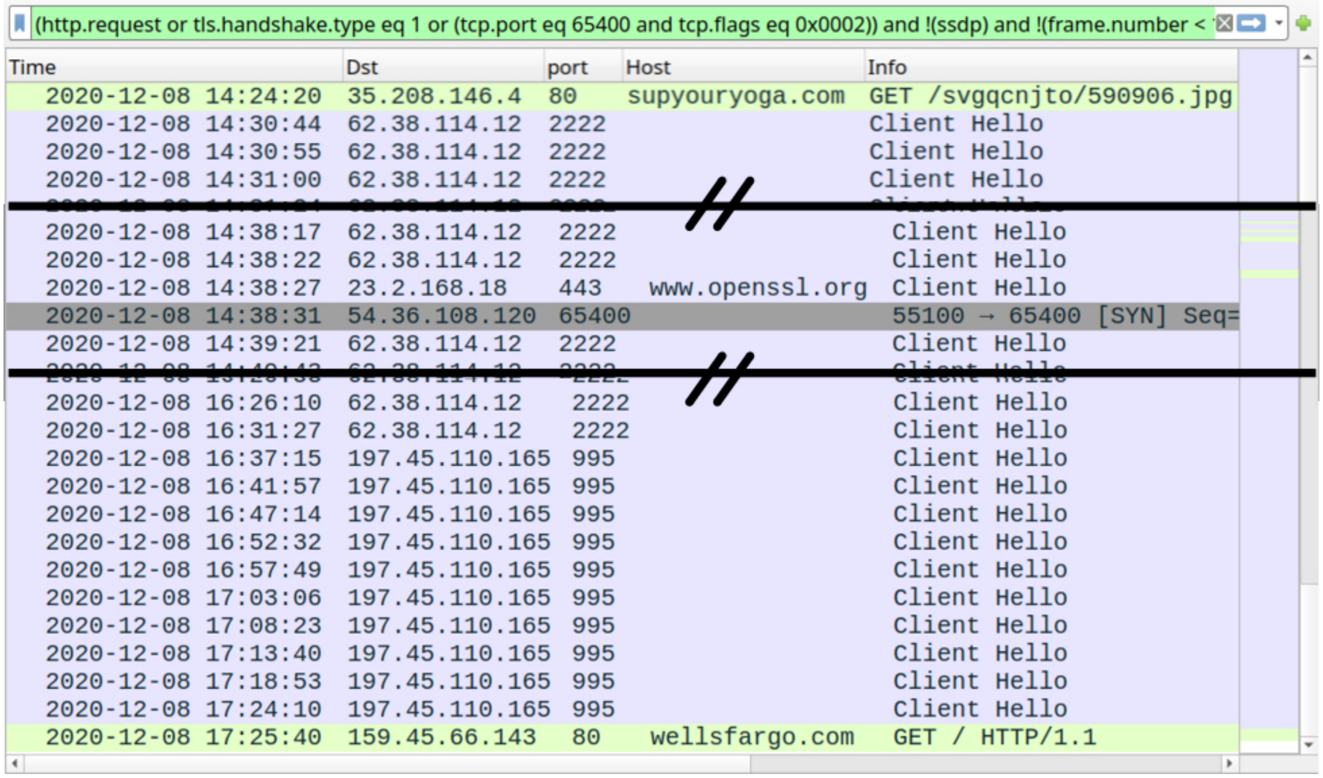
Show and save data as: ASCII

Stream: 31

Find:

Shown above: HTTP traffic that returned a Windows DLL file for Qakbot.

Filtering the traffic in Wireshark, we find typical Qakbot post-infection activity. But approximately 3 hours after the initial infection, I also saw web traffic to ***wellsfargo[.]com***, which was unusual--especially since no browser had opened on the desktop of the infected Windows host.

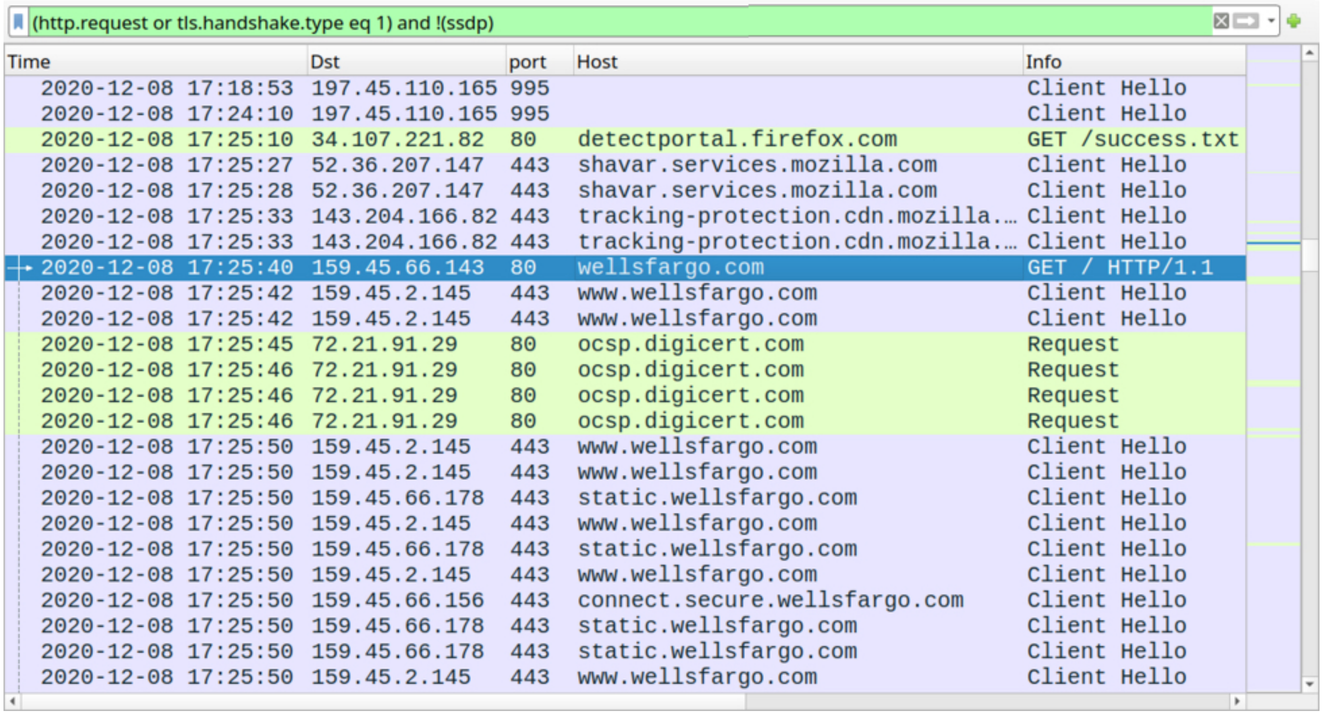


(http.request or tls.handshake.type eq 1 or (tcp.port eq 65400 and tcp.flags eq 0x0002)) and !(ssdp) and !(frame.number <

Time	Dst	port	Host	Info
2020-12-08 14:24:20	35.208.146.4	80	supyouryoga.com	GET /svgqcnjto/590906.jpg
2020-12-08 14:30:44	62.38.114.12	2222		Client Hello
2020-12-08 14:30:55	62.38.114.12	2222		Client Hello
2020-12-08 14:31:00	62.38.114.12	2222		Client Hello
2020-12-08 14:38:17	62.38.114.12	2222		Client Hello
2020-12-08 14:38:22	62.38.114.12	2222		Client Hello
2020-12-08 14:38:27	23.2.168.18	443	www.openssl.org	Client Hello
2020-12-08 14:38:31	54.36.108.120	65400		55100 → 65400 [SYN] Seq=
2020-12-08 14:39:21	62.38.114.12	2222		Client Hello
2020-12-08 16:26:10	62.38.114.12	2222		Client Hello
2020-12-08 16:31:27	62.38.114.12	2222		Client Hello
2020-12-08 16:37:15	197.45.110.165	995		Client Hello
2020-12-08 16:41:57	197.45.110.165	995		Client Hello
2020-12-08 16:47:14	197.45.110.165	995		Client Hello
2020-12-08 16:52:32	197.45.110.165	995		Client Hello
2020-12-08 16:57:49	197.45.110.165	995		Client Hello
2020-12-08 17:03:06	197.45.110.165	995		Client Hello
2020-12-08 17:08:23	197.45.110.165	995		Client Hello
2020-12-08 17:13:40	197.45.110.165	995		Client Hello
2020-12-08 17:18:53	197.45.110.165	995		Client Hello
2020-12-08 17:24:10	197.45.110.165	995		Client Hello
2020-12-08 17:25:40	159.45.66.143	80	wellsfargo.com	GET / HTTP/1.1

Shown above: Traffic from the infection filtered in Wireshark.

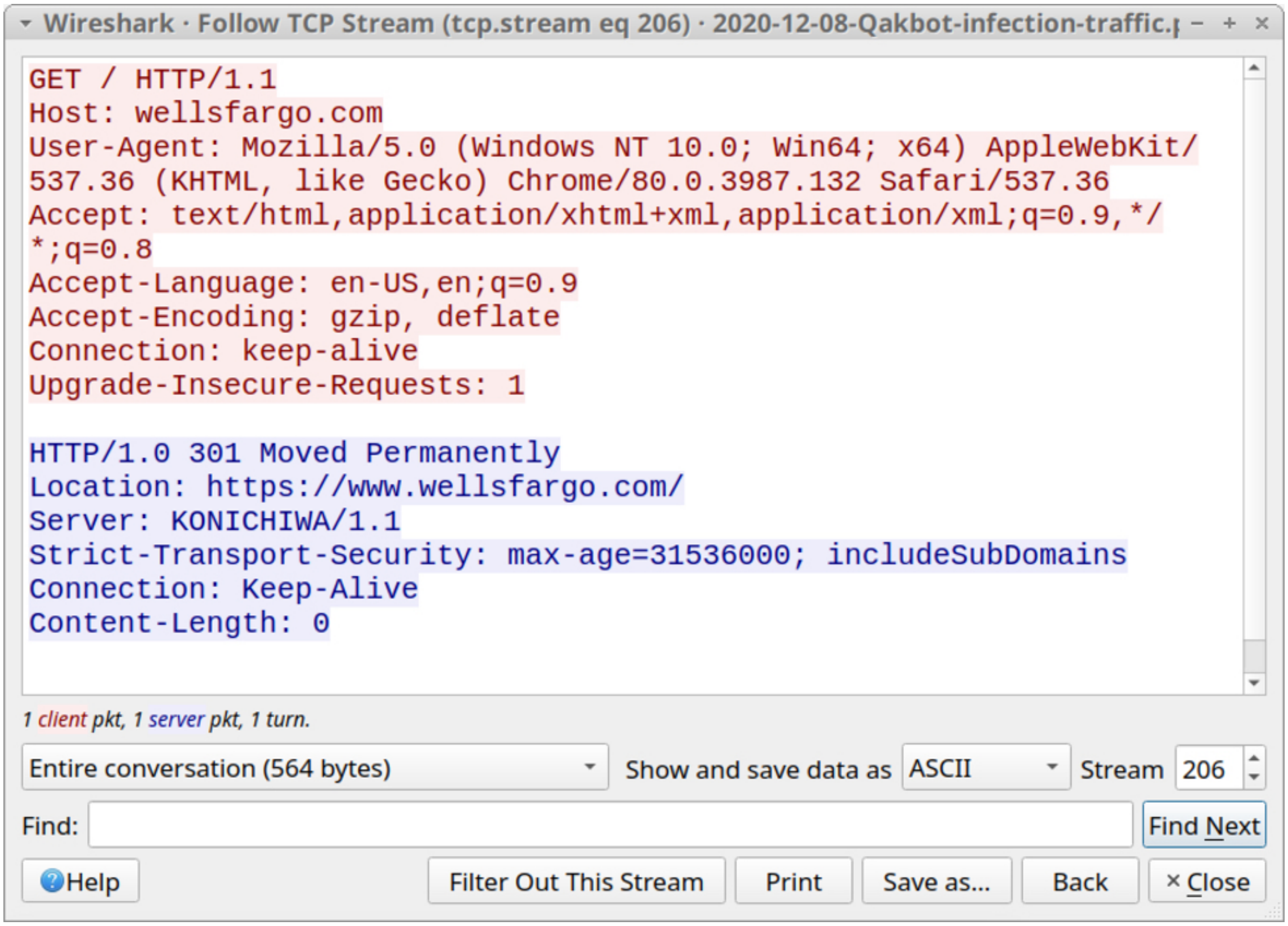
The user-agent string in HTTP traffic to ***wellsfargo[.]com*** indicated it may have been caused by Google Chrome. Keep in mind the user-agent string is often spoofed during malware infections. I also saw web traffic associated with the Firefox web browser. This traffic is likely related to one of the Qakbot modules; however, I could not find any modules saved to disk on my infected host.



(http.request or tls.handshake.type eq 1) and !(ssdp)

Time	Dst	port	Host	Info
2020-12-08 17:18:53	197.45.110.165	995		Client Hello
2020-12-08 17:24:10	197.45.110.165	995		Client Hello
2020-12-08 17:25:10	34.107.221.82	80	detectportal.firefox.com	GET /success.txt
2020-12-08 17:25:27	52.36.207.147	443	shavar.services.mozilla.com	Client Hello
2020-12-08 17:25:28	52.36.207.147	443	shavar.services.mozilla.com	Client Hello
2020-12-08 17:25:33	143.204.166.82	443	tracking-protection.cdn.mozilla.…	Client Hello
2020-12-08 17:25:33	143.204.166.82	443	tracking-protection.cdn.mozilla.…	Client Hello
2020-12-08 17:25:40	159.45.66.143	80	wellsfargo.com	GET / HTTP/1.1
2020-12-08 17:25:42	159.45.2.145	443	www.wellsfargo.com	Client Hello
2020-12-08 17:25:42	159.45.2.145	443	www.wellsfargo.com	Client Hello
2020-12-08 17:25:45	72.21.91.29	80	ocsp.digicert.com	Request
2020-12-08 17:25:46	72.21.91.29	80	ocsp.digicert.com	Request
2020-12-08 17:25:46	72.21.91.29	80	ocsp.digicert.com	Request
2020-12-08 17:25:46	72.21.91.29	80	ocsp.digicert.com	Request
2020-12-08 17:25:50	159.45.2.145	443	www.wellsfargo.com	Client Hello
2020-12-08 17:25:50	159.45.2.145	443	www.wellsfargo.com	Client Hello
2020-12-08 17:25:50	159.45.66.178	443	static.wellsfargo.com	Client Hello
2020-12-08 17:25:50	159.45.2.145	443	www.wellsfargo.com	Client Hello
2020-12-08 17:25:50	159.45.66.178	443	static.wellsfargo.com	Client Hello
2020-12-08 17:25:50	159.45.2.145	443	www.wellsfargo.com	Client Hello
2020-12-08 17:25:50	159.45.66.156	443	connect.secure.wellsfargo.com	Client Hello
2020-12-08 17:25:50	159.45.66.178	443	static.wellsfargo.com	Client Hello
2020-12-08 17:25:50	159.45.66.178	443	static.wellsfargo.com	Client Hello
2020-12-08 17:25:50	159.45.2.145	443	www.wellsfargo.com	Client Hello

Shown above: Filtering the traffic in Wireshark to show Firefox traffic, and other web traffic to wellsfargo[.]com from the infected host.



Wireshark · Follow TCP Stream (tcp.stream eq 206) · 2020-12-08-Qakbot-infection-traffic.p

```
GET / HTTP/1.1
Host: wellsfargo.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/
537.36 (KHTML, like Gecko) Chrome/80.0.3987.132 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/
*;q=0.8
Accept-Language: en-US,en;q=0.9
Accept-Encoding: gzip, deflate
Connection: keep-alive
Upgrade-Insecure-Requests: 1

HTTP/1.0 301 Moved Permanently
Location: https://www.wellsfargo.com/
Server: KONICHIWA/1.1
Strict-Transport-Security: max-age=31536000; includeSubDomains
Connection: Keep-Alive
Content-Length: 0
```

1 client pkt, 1 server pkt, 1 turn.

Entire conversation (564 bytes)

Show and save data as: ASCII

Stream: 206

Find:

Shown above: Traffic to wellsfargo[.]com appears to be from Chrome, if the user-agent string is correct.

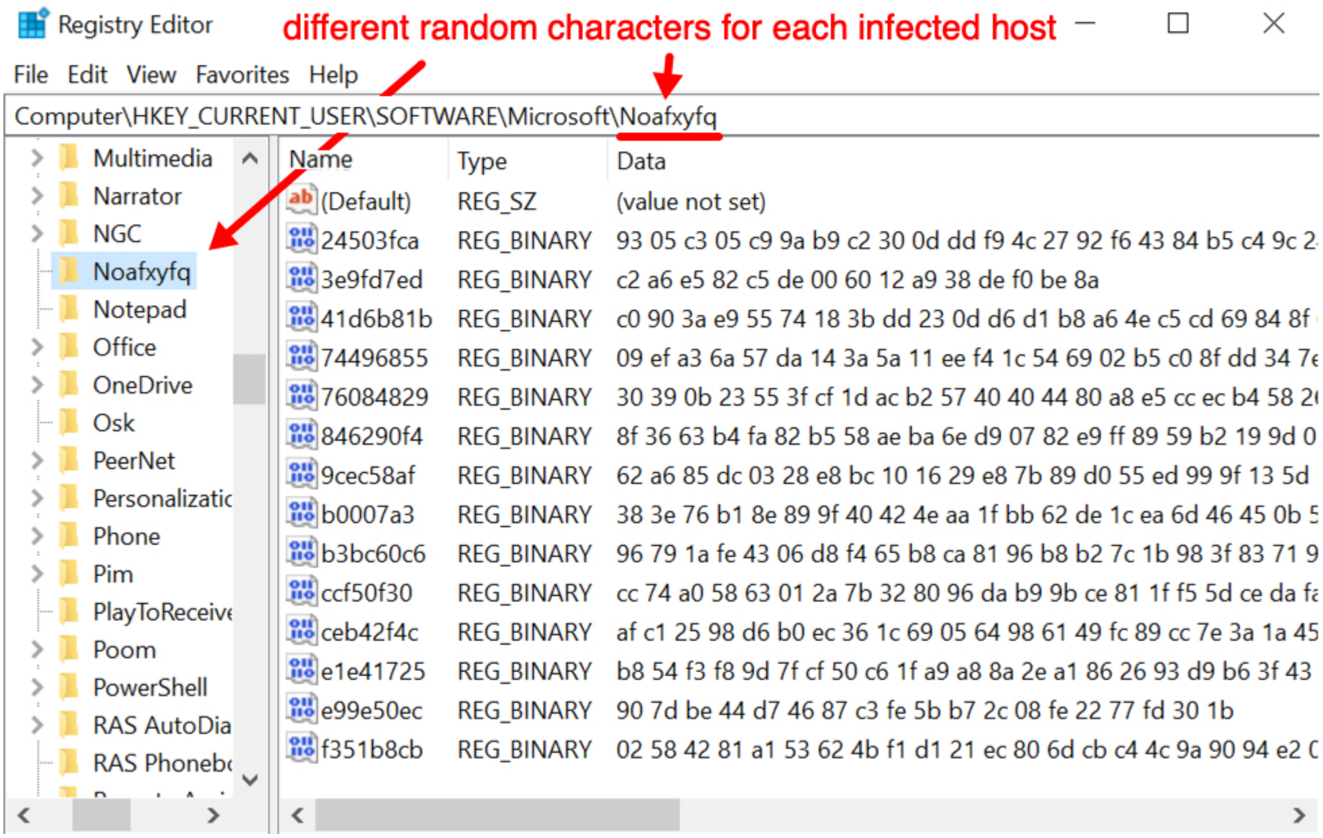
Qakbot malware version update

Sometime in late-November 2020, Qakbot malware was updated. I know of at least 3 related things that are noticeably different than before.

1) The Qakbot binary retrieved by Microsoft Office macros changed from an EXE to a DLL.

- Prior to the update, the initial Qakbot binary was an EXE made persistent through a Windows registry update at ***HKCU\SOFTWARE\Microsoft\WIndows\CurrentVersion\Run***.
- After the update, the initial Qakbot binary has been a DLL file, and there is no longer a Windows registry update at ***HKCU\SOFTWARE\Microsoft\WIndows\CurrentVersion\Run***.

2) Qakbot now creates other Windows registry updates. These updates are located at ***HKCU\SOFTWARE\Microsoft*** under a key that consists of a unique alphabetical string for each infected host. It consists of several entries containing encoded binary data as shown in the example below.



different random characters for each infected host

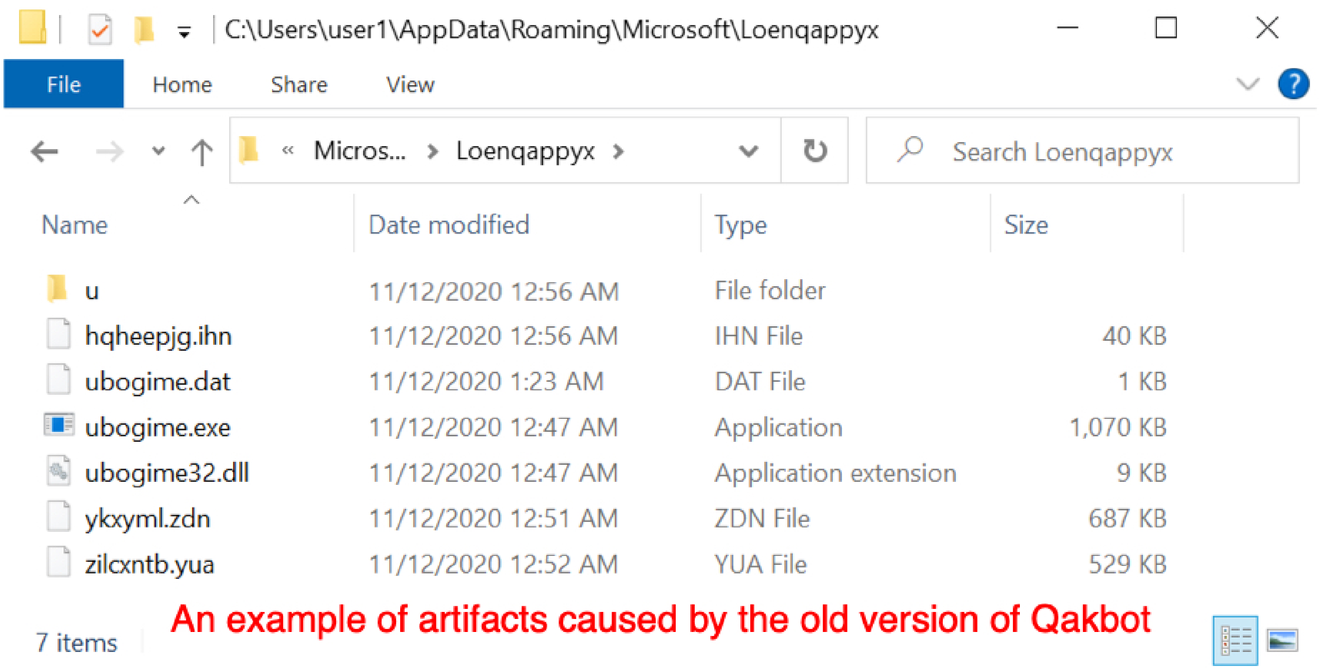
Computer\HKEY_CURRENT_USER\SOFTWARE\Microsoft\Noafxyfq

Name	Type	Data
(Default)	REG_SZ	(value not set)
24503fca	REG_BINARY	93 05 c3 05 c9 9a b9 c2 30 0d dd f9 4c 27 92 f6 43 84 b5 c4 9c 2
3e9fd7ed	REG_BINARY	c2 a6 e5 82 c5 de 00 60 12 a9 38 de f0 be 8a
41d6b81b	REG_BINARY	c0 90 3a e9 55 74 18 3b dd 23 0d d6 d1 b8 a6 4e c5 cd 69 84 8f
74496855	REG_BINARY	09 ef a3 6a 57 da 14 3a 5a 11 ee f4 1c 54 69 02 b5 c0 8f dd 34 7e
76084829	REG_BINARY	30 39 0b 23 55 3f cf 1d ac b2 57 40 40 44 80 a8 e5 cc ec b4 58 2
846290f4	REG_BINARY	8f 36 63 b4 fa 82 b5 58 ae ba 6e d9 07 82 e9 ff 89 59 b2 19 9d 0
9cec58af	REG_BINARY	62 a6 85 dc 03 28 e8 bc 10 16 29 e8 7b 89 d0 55 ed 99 9f 13 5d
b0007a3	REG_BINARY	38 3e 76 b1 8e 89 9f 40 42 4e aa 1f bb 62 de 1c ea 6d 46 45 0b 5
b3bc60c6	REG_BINARY	96 79 1a fe 43 06 d8 f4 65 b8 ca 81 96 b8 b2 7c 1b 98 3f 83 71 9
ccf50f30	REG_BINARY	cc 74 a0 58 63 01 2a 7b 32 80 96 da b9 9b ce 81 1f f5 5d ce da f
ceb42f4c	REG_BINARY	af c1 25 98 d6 b0 ec 36 1c 69 05 64 98 61 49 fc 89 cc 7e 3a 1a 45
e1e41725	REG_BINARY	b8 54 f3 f8 9d 7f cf 50 c6 1f a9 a8 8a 2e a1 86 26 93 d9 b6 3f 43
e99e50ec	REG_BINARY	90 7d be 44 d7 46 87 c3 fe 5b b7 2c 08 fe 22 77 fd 30 1b
f351b8cb	REG_BINARY	02 58 42 81 a1 53 62 4b f1 d1 21 ec 80 6d cb c4 4c 9a 90 94 e2 0

Shown above: An example of Windows registry update caused by the newest version of Qakbot.

3) The directory for Qakbot artifacts under ***C:\Users\[username]\AppData\Roaming\Microsoft*** now has fewer files. Before the version update, we saw a Windows EXE for Qakbot in this directory, and it was kept persistent in the

Windows registry (see item 1 above). Now, the folder no longer has an EXE and some other files are missing. Compare the two images below.

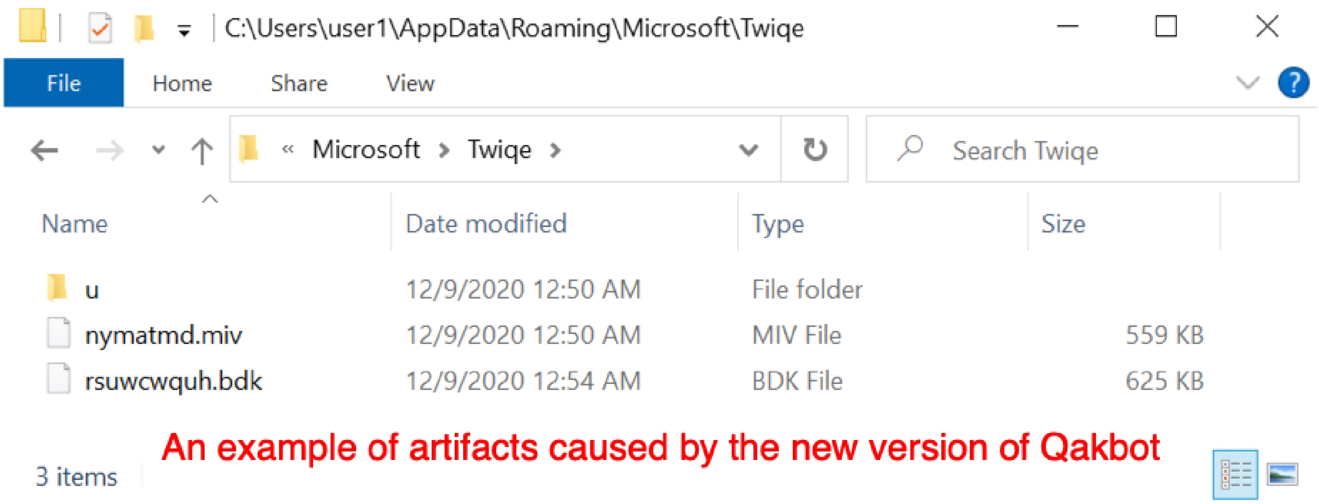


C:\Users\user1\AppData\Roaming\Microsoft\Loenqappyx

Name	Date modified	Type	Size
u	11/12/2020 12:56 AM	File folder	
hqheepjg.ihn	11/12/2020 12:56 AM	IHN File	40 KB
ubogime.dat	11/12/2020 1:23 AM	DAT File	1 KB
ubogime.exe	11/12/2020 12:47 AM	Application	1,070 KB
ubogime32.dll	11/12/2020 12:47 AM	Application extension	9 KB
ykxyml.zdn	11/12/2020 12:51 AM	ZDN File	687 KB
zilcxntb.yua	11/12/2020 12:52 AM	YUA File	529 KB

An example of artifacts caused by the old version of Qakbot

Shown above: An example of artifact caused by the old version of Qakbot.



C:\Users\user1\AppData\Roaming\Microsoft\Twiqe

Name	Date modified	Type	Size
u	12/9/2020 12:50 AM	File folder	
nymatmd.miv	12/9/2020 12:50 AM	MIV File	559 KB
rsuwcwquh.bdk	12/9/2020 12:54 AM	BDK File	625 KB

An example of artifacts caused by the new version of Qakbot

Shown above: Artifacts stores to the same type of directory after the late-November 2020 version update of Qakbot.

Qakbot's version update has resulted in other characteristics of the malware, and I'm certain someone will publish a more detailed write-up about it. These three changes are the one's I've noticed, but I focus mostly on dynamic analysis (not code analysis or reverse engineering).

Indicators of Compromise (IoC)

The following are IoCs from my Qakbot infection from Tuesday 2020-12-08.

ZIP archives from 4 malspam examples:

- 2ccc14f2bab2e9eb1d7228e225afda558fd4b52ed670303a912ace1984b35b06
  Document_1204350147-Copy.zip
- fa9935e6cda06866cb5aa062c16a73fdc85bd4146dca67202d22e225ddd3193b
  Document_1356928040-Copy.zip
- 0a3a6163a5e8e372fa96efbef3feb793463f4e39bd2c4d6ea03afce045f90636
  Document_1495694596-Copy.zip
- 66036cf566386c159e49191125497c77c13c75778492519000b9f61a4afdedad
  Document_501487929-Copy.zip

Excel spreadsheets extracted from the above ZIP archives:

- adad807fa22f398e0a40396ed65d0827f9f14baf7e1281b713dfb17e2683d743
  Document_1204350147-Copy.xls
- e14f6ab34e3506d6985816af85935932fb6faf8bad9d2c7dd96d6011d7c21a33
  Document_1356928040-Copy.xls
- 4e2f37d4228e78faa1f34121ee934f58e1a9862ad6f183edf4c24e08cda20363
  Document_1495694596-Copy.xls
- 94d759f43bcc647f7233e19ddc160a6b43458dcde6d2ea4274c8c06b2890def2
  Document_501487929-Copy.xls

HTTP traffic after enabling that returned a Qakbot DLL file:

> 35.208.146[.]4 port 80 - ***supyouryoga[.]com*** - GET /svgqcnjto/590906.jpg

Qakbot post-infection traffic:

- 62.38.114[.]12 port 2222 - HTTPS traffic caused by Qakbot
- 197.45.110[.]165 port 995 - HTTPS traffic caused by Qakbot
- port 443 - ***www.openssl[.]org*** - connectivity check caused by Qakbot
- 54.36.108[.]120 port 65400 - TCP traffic caused by Qakbot

Unusual (to me) activity from Qakbot-infected host:

- port 80 - ***wellsfargo[.]com*** - GET /
- various IP addresses over TCP port 443 - Wells Fargo-related domains - traffic caused by viewing *wellsfargo[.[com*
- *Firefox-related HTTP and HTTPS web traffic*

Malware from an infected Windows host:

- SHA256 hash:
  5060806228d3f2c1afd09566d0d2fa6b2e56f844cd044c4c4e6e7ade9fef3a22
- File size: 350,928 bytes

- File retrieved from: hxxp://supyouryoga[.]com/svgqcnjto/590906.jpg
- File saved to victim as: C:\Users\*[username]*\AppData\Kipofe.mmaallaauu
- File description: DLL file for Qakbot retrieved by macro from Document_1495694596-Copy.xls
- Run method: Rundll32.exe *[filename]*,DllRegisterServer

Final words

Qakbot been active for several years, and it continues to evolve. The latest version update has some significant changes, but infection traffic on vulnerable Windows hosts remains similar to what we've seen before with Qakbot.

A pcap of the infection traffic reviewed in this dairy and 4 examples of Qakbot malspam are available here.

Brad Duncan
brad [at] malware-traffic-analysis.net

Keywords: Excel macros malspam Qakbot Qbot

0 comment(s)

DEV522
Defending Web Application Security Essentials
LEARN MORE
Learn to defend your apps before they're hacked

Top of page
×

Diary Archives