Detecting OneNote (.One) Malware Delivery

I opened a dozen malicious OneNote files and clicked on every link so you don't have to

Micah Babinski · Follow

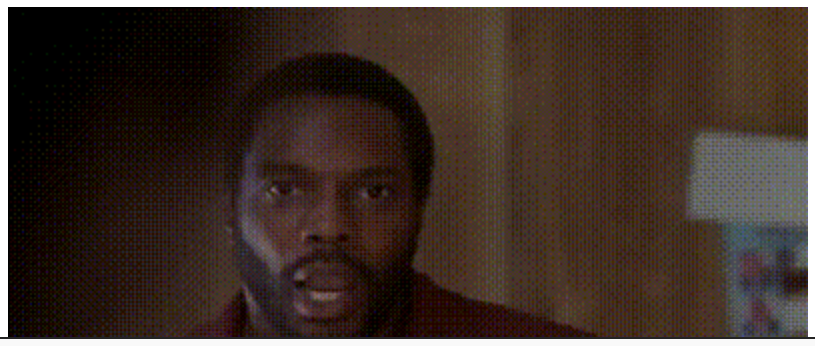
9 min read · Jan 31, 2023

My future in graphic design is bright (everyone says so).

As tweets like the one above gradually seeped into my consciousness, I de[redacted]es th[redacted] victim networks. I'd really enjoyed the QakBot/IcedID/HTML smuggling research I did at the end of 2022 and wanted to see if I could repeat this process for OneNote-delivered malware, for the educational benefit and [mild] enjoyment of all. My objectives were:

1. 💡 Understand how OneNote is used to deliver malware.
2. 🔭 Observe OneNote malware delivery in my lab.
3. 📖 Review existing log-based detections for this activity, and identify possible ways to augment or strengthen these.
4. ✍️ Write and share new or improved rules to detect OneNote malware delivery.
5. 🍪 Celebrate with a tasty treat.

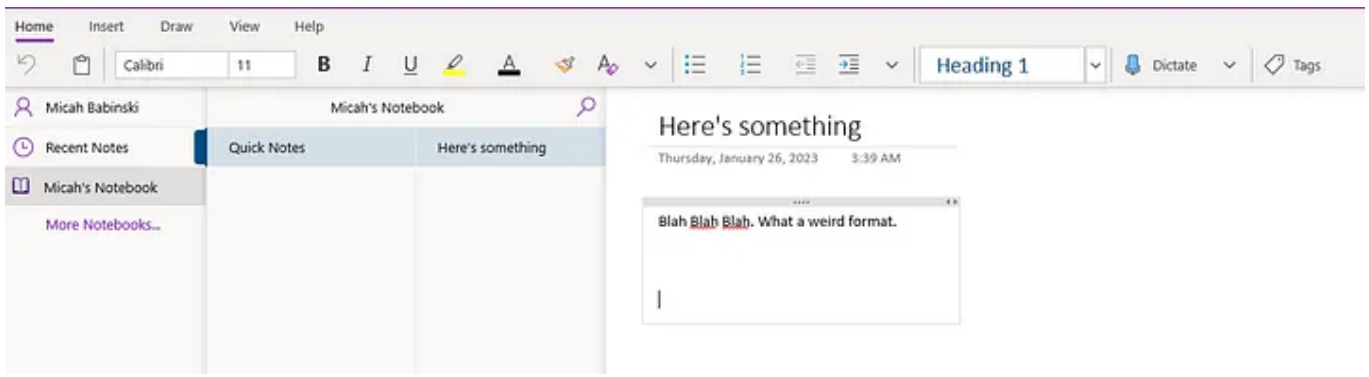
I hope this article provides another useful example of how current or aspiring detection specialists can research adversary techniques, extract observable patterns, and design/share detection rules to put those attackers in our sights!



The Simpsons Covered OneNote Malware Already

OneNote About OneNote

Before getting started, I had to acknowledge that I didn't really know much of anything about OneNote. Ok, I know it's Microsoft's built-in note-taking app. I had explored it briefly during my last job as a Security Analyst but gave it up in favor of good old pen and paper (seriously). So, in the interest of forming some very baseline familiarity with the tool, I fired it up in my lab and created a very simple Notebook, as shown below:



Behold

But what about a .one file?

With my example Notebook in the bag, I wanted to understand what a .one file is, how you create one, how you use one, and how it is being abused by threat actors. After researching a bit online, I found out that you can export a OneNote notebook to a .one file that can in turn be imported into the OneNote collection of another user so that they have their own copy of the notebook. This is a little odd, as OneNote notebooks are definitely intended to be shared directly via the web, but I suppose it's conceivable that someone wanting to use or share notebooks in a disconnected/offline

4. The malicious HTA or VBS file calls the WMI provider host

5. Simultaneously to the step above, CDM/PowerShell are used to retrieve and open a legitimate .one file template from onenotegem.com, a site with helpful OneNote templates, or a compromised website. This leads the victim to think they have what they need, making them less likely to report the infection to IT/Security.
6. The malicious batch file from step 4 copies the PowerShell executable and uses it to run an encrypted payload, which is the AsyncRAT trojan or similar info-stealing malware.

With this basic understanding in mind, it was time to gather some sample malicious .one files and test them out in the lab!

Gathering Samples

To gather my samples I turned to Malware Bazaar, an excellent resource I mentioned in my last post about HTML smuggling. I found I could efficiently find and download these samples using the tags “one” and “OneNote:”

https://bazaar.abuse.ch/browse/tag/OneNote/

https://bazaar.abuse.ch/browse/tag/one/

As before, I downloaded and extracted a number of these samples, naming them according to the “humanhash” property of each sample:

CW: Extremely Weird

Testing and Observing OneNote Malware Delivery

The sample notebooks I tested all contained some variation of this look and feel:

I d[redacted]t could harm my computer. After a few seconds, an invoice from Excel Business Systems pops up in OneNote, and a command prompt window appears for a split second. Below, I've included an annotated screenshot of the Process Creation logs, which I believe will be useful for understanding the chronology of the events:

And here, we can see additional log details from the final step in the malware delivery process, showing us based on the OriginalFileName property that system32.bat.exe is in fact a copied version of Microsoft PowerShell.exe:

Why you gotta be that way, RegAsm.exe? What did I do to you?

Significantly, this malicious powershell script is still available on transfer.sh!

VirusTotal

VirusTotal

VirusTotalwww.virustotal.com

Another strange and possibly-relevant observation was that, in the screenshot above of the OneNote notebook with the row of malicious WSF attachments, the filenames contained some sort of non-printable character in the filename which causes the filename to appear partially-reversed in the logs:

Record scratch noise

Masquerading: Right-to-Left Override

attack.mitre.org

Existing Detections

Of the existing log-based detection rules I found, the best one is "Suspicious OneNote Child Process":

sigma/proc_creation_win_susp_microsoft_onenote_child_process.yml at master · SigmaHQ/sigma

Main Rule Repository. Contribute to SigmaHQ/sigma development by creating an account on GitHub.

github.com

This one rule is really effective on its own! It looks for suspicious processes spawned by OneNote.exe, including all of the ones that I observed while executing my OneNote samples.

New Detection Ideas

I wondered, however, if I could come up with some additional rules which would match on the activity I observed, including the subtle variations. After all, building up detection "in depth" could result in multiple alerts, which could more reliably point a SOC to the correct investigative pathway. My ideas included:

- Double extension Image (process name). This could apply to process

You can see the recipe right here: https://regex101.com/r/zLgqzk/1. I dropped it into the following Sigma rule which converted successfully into a Splunk query that detected all of the malicious OneNote attachments in my process creation logs. Note: the Regex below *looks* erroneous, but that's only because the invisible unicode character reversed the character order. This rule really works!

```yaml
title: Suspicious Command Line Containing Right-to-Left Override
id: ad691d92-15f2-4181-9aa4-723c74f9ddc3
status: experimental
description: Detects the presence of the u202+E character, which causes a terminal,
references:
    - https://redcanary.com/blog/right-to-left-override/
    - https://unicode-explorer.com/c/202E
author: Micah Babinski, @micahbabinski
date: 2023/01/30
tags:
    - attack.defense_evasion
    - attack.t1036
    - attack.t1036.002
logsource:
    category: process_creation
    product: windows
detection:
    selection:
        # you can't see it, but trust me, there's a right-to-left override character
        CommandLine|re: ^.*$*.
    condition: selection
falsepositives:
    - Unknown
level: high
```

And now, with my fifth objective in mind, I am off to eat a cookie, or should it

That's all for now! As always, Happy Analyzing! 🧐

Cybersecurity | Information Security | Malware | Detection Engineering | Threat Intelligence

Written by Micah Babinski

Follow

257 Followers

Cybersecurity pro, featuring bagpiping and GIS chops. Lives with wife Quinn and son Malcolm. Loves mountains, Indian food, and mountains of Indian food.

Micah Babinski

Cr
(a

A few months ago I decided to check and see whether there was a Sigma backend for...

Apr 12, 2022 21 2

Micah Babinski

Dec 28, 2022 23 1

See all from Micah Babinski

Recommended from Medium

Aardvark Infinity in Aardvark Infinity

Set Up a Windows 11 Malware Analysis Lab for Reverse...

Aug 29 12

Shereen

Leveraging Caldera for Adversary Emulation

C2 Simulation using Sandcat agent on Linux Target

Jul 31

Nathan Hueck

KC
id

We can begin with comprehensive approach to identifying executable files in the Window...

Jun 7

Rodolfo Santos Flaborea

which a previous post has already...

Oct 9

CICADA8

Hijack the TypeLib. New COM persistence technique

A new way of persistence on Windows systems via COM. Down with COM Hijacking...

Oct 22 24

marianita_cloud

AWS Cloud Red Team Specialist [CARTS] Exam Review

Hello Cloud Security community, I am pleased to share my experience and process for...

Sep 2 2 2

See more recommendations