

SECTRIO

MALWARE REPORT



RevengeRAT

Date: 24/06/2021

Shikha Sangwan

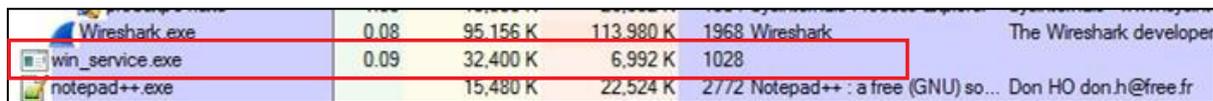
RevengeRAT is a potential Remote-Access Trojan malware. Microsoft has issued an alert over a RAT dubbed RevengeRAT that it says has been used to target Aerospace and Travel sectors with spear-phishing emails. This RAT is used as a Crypter-as-a-Service.

Overview

This RAT once installed into the system, checks for any anti-virus or any security software in the system and drops multiple executables in the system for persistence. The dropped malware is a backdoor which looks like a genuine application. The malware connects to the C&C server and steals sensitive information about the system, physical memory, webcams, and the machine's user. It encodes the collected information, sends it to the C2 server and compromise the victim's machine.

Technical Analysis

Once we run the malware in the system, it drops and starts a program with name "win_service.exe" which is a backdoor, and it misleads the user as its name looks like a legitimate software.

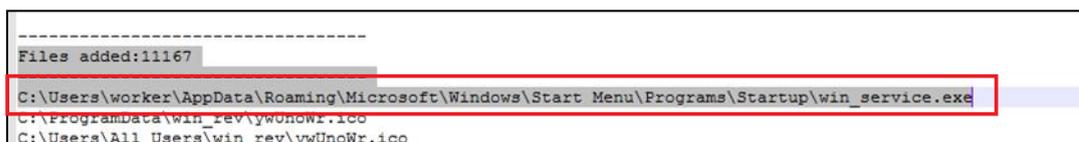


Wireshark.exe	0.08	95,156 K	113,980 K	1968	Wireshark	The Wireshark developer
win_service.exe	0.09	32,400 K	6,992 K	1028		
notepad++.exe		15,480 K	22,524 K	2772	Notepad++ : a free (GNU) so...	Don HO don.h@free.fr

Figure 1 Trojan Backdoor

Persistence

It drops the PE file to the start-up folder (as shown in the figure below). Malware uses start-up items to automatically execute at boot to establish persistence. It also creates a start menu entry and stores files in the Windows start menu directory.

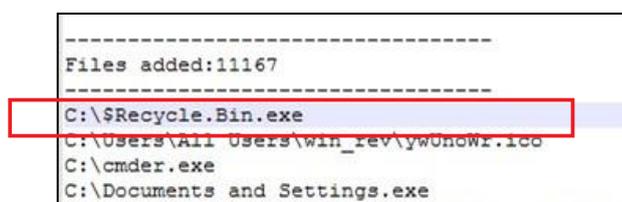


```
-----  
Files added:11167  
C:\Users\worker\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup\win_service.exe  
C:\ProgramData\win_rev\ywOnoWr.ico  
C:\Users\All Users\win_rev\ywOnoWr.ico
```

Figure 2 Dropped PE file in the start-up folder

Hooking

It creates files in the recycle bin and volume driver for hiding and protection.



```
-----  
Files added:11167  
-----  
C:\$Recycle.Bin.exe  
C:\Users\All Users\win_rev\ywOnoWr.ico  
C:\cmdr.exe  
C:\Documents and Settings.exe
```

Figure 3 File created in the recycle bin

It turns off the Windows Error reporting for the current user, as shown below in the disassembly at address "0000000076F28FE6". This disables application error reporting messages to hide itself.

```

0000000076F28FE6 48 89 mov qword ptr ss:[rsp+
0000000076F28FE3 48 88 mov rbx,rcx
0000000076F28FE0 48 8D lea rdx,qword ptr ds:[0000000076FAEBD0:L"\\Registry\\Machine\\Software\\Microsoft\\Windows\\Windows Error Reporting\\W
0000000076F28FE5 48 89 mov qword ptr ss:[rs
0000000076F28FF7 FF 15 call qword ptr ds:[<6A
0000000076F28FFD 33 FF xor edi,edi
0000000076F28FFF 48 8D lea rax,qword ptr ss:[
0000000076F29004 4C 8D lea r8,qword ptr ss:[
0000000076F29009 8D 57 lea edx,qword ptr ds:[
0000000076F2900C 48 8B mov rcx,rbx
0000000076F2900F C7 44 mov dword ptr ss:[rsp+30:'0'
0000000076F29017 48 89 mov qword ptr ss:[rsp+
0000000076F2901C C7 44 mov dword ptr ss:[rsp+40:'e'
0000000076F29024 48 89 mov qword ptr ss:[rsp+
0000000076F29029 48 89 mov qword ptr ss:[rsp+
0000000076F2902E 48 89 mov qword ptr ss:[rsp+
0000000076F29033 FF 15 call qword ptr ds:[<62
0000000076F29039 85 C0 test eax,eax
0000000076F2903B 0F 85 jne kernel32.76F3A9C7
0000000076F29041 8D 47 lea eax,qword ptr ds:[
0000000076F29044 48 8B mov rbx,qword ptr ss:[
0000000076F29049 48 8B mov rdi,qword ptr ss:[
0000000076F2904E 48 83 add rsp,65

```

Figure 4 Address: 0000000076F28FE6 disabling application error reporting

Backdoor

It reads .ini files from the desktop to get the configuration information of different programs that runs in the system.

```

-----
Files accessed:1767
-----
C:\Users\worker\Desktop\desktop.ini
-----

```

Figure 5 Read desktop's ini file

Then the backdoor steals critical system information like the victim's user name and computer name, Windows system information, capacity of the physical memory, type of security software installed in the system, and language used on the victim machine. (As shown in the disassembly below)

```

0000000076F1E562 48 85 test rax,rax
0000000076F1E565 0F 85 jne kernel32.76F3D339
0000000076F1E568 48 8D lea rdx,qword ptr ds:[76FAD0:0000000076FAE40:L"\\Registry\\Machine\\System\\CurrentControlSet\\Control\\ComputerName"
0000000076F1E572 48 8D lea rcx,qword ptr ss:[rsp+5
0000000076F1E577 FF 15 call qword ptr ds:[<6A7C11ni
0000000076F1E57D 48 8D lea rax,qword ptr ss:[rsp+5
0000000076F1E582 4C 8D lea r8,qword ptr ss:[rsp+80
0000000076F1E58A 48 8D lea rcx,qword ptr ss:[rsp+4
0000000076F1E58F BA 19 mov edx,20019
0000000076F1E594 48 89 mov qword ptr ss:[rsp+90,'r
0000000076F1E59C C7 84 mov dword ptr ss:[rsp+80,'3
0000000076F1E5A7 4C 89 mov qword ptr ss:[rsp+88,'r
0000000076F1E5AF C7 84 mov dword ptr ss:[rsp+96,'4
0000000076F1E5B4 4C 89 mov qword ptr ss:[rsp+A0,'r
0000000076F1E5C2 4C 89 mov qword ptr ss:[rsp+A8,'r
0000000076F1E5CA FF 15 call qword ptr ds:[<62W0pen
0000000076F1E5D0 3D 34 cmp eax,C0000034
0000000076F1E5D5 0F 84 je kernel32.76F3D35D
0000000076F1E5DB 85 C0 test eax,eax
0000000076F1E5DD 0F 85 jne kernel32.76F48E60

```

Figure 7 address:0000000076F1E568 reading computer's name

```

0000000076F72AB6 E8 6D call <kernel32.FreeLibrary>
0000000076F72ABB EB D8 jmp kernel32.76F72A95
0000000076F72ABD 48 8D lea rdx,qword ptr ds:[76FC81F0:0000000076FC81F0:"GetUserName"
0000000076F72AC4 E8 CE call <kernel32.GetProcAddress>
0000000076F72AC9 48 8B mov rcx,rbx
0000000076F72ACC 4C 8B mov r12,rax
0000000076F72ACF 48 3B cmp rax,rdi
0000000076F72AD2 74 E2 je kernel32.76F72AD4
0000000076F72AD4 48 8D lea rdx,qword ptr ds:[77004190:0000000077004190:"ImpersonateNamedPipeClient"
0000000076F72ADB E8 B8 call <kernel32.GetProcAddress>
0000000076F72AE0 4C 8B mov r13,rax
0000000076F72AE3 48 3B cmp rax,rdi
0000000076F72AE6 75 05 jne kernel32.76F72AED
0000000076F72AE8 48 8B mov rcx,rbx
0000000076F72AEB EB C9 jmp kernel32.76F72AB6
0000000076F72AED 4C 8D lea r9,qword ptr ss:[rsp+78]
0000000076F72AF2 41 B0 mov r8b,1
0000000076F72AF5 BA 04 mov edx,4
0000000076F72AFA 48 C7 mov rcx,FFFFFFFFFFFFFFFF

```

Figure 6 address:0000000076F72ABD reading user's name

0000000071171A5	42 54	push r12	
0000000071171A7	48 81	sub rsp,c0	
0000000071171AE	48 88	mov rbx,rcx	
0000000071171B1	48 8D	lea rdx,qword ptr ds:[77117490]	000000007117490:L"System volume information"
0000000071171B8	48 8D	lea rcx,qword ptr ss:[rsp+70]	
0000000071171B0	E8 8E	call <ntdll.RtlInitUnicodeString>	
0000000071171C2	0F 87	movzx edx,word ptr ds:[rbx]	
0000000071171C5	0F 87	movzx edi,word ptr ss:[rsp+70]	
0000000071171CA	66 03	add dx,di	
0000000071171CD	66 89	mov word ptr ss:[rsp+60],dx	
0000000071171D2	66 3B	cmp dx,word ptr ds:[rbx]	
0000000071171D5	0F 82	jb ntdll.77117470	
0000000071171D8	66 3B	cmp dx,di	
0000000071171DE	0F 82	jb ntdll.77117470	
0000000071171E4	0F 87	movzx ecx,word ptr ds:[rbx]	
0000000071171E7	48 8B	mov rax,qword ptr ds:[rbx+8]	
0000000071171EB	33 ED	xor ebp,ebp	
0000000071171ED	44 8D	lea r12d,qword ptr ss:[rbp+5c]	
0000000071171F1	48 D1	shr rcx,1	
0000000071171F4	66 44	cmp word ptr ds:[rax+rcx*2-2],r12	
0000000071171FA	74 0E	je ntdll.7711720A	
0000000071171FC	66 83	add dx,2	
000000007117200	40 86	mov sil,1	

Figure 8 address:0000000071171B1 getting system's volume information

0000000076F72C6D	83 F8	cmp eax,2	
0000000076F72C70	75 3E	jne kernel32.76F72CB0	
0000000076F72C77	48 8D	lea rdx,qword ptr ds:[77004180]	0000000077004180:"GetMemory"
0000000076F72C7E	8D 48	lea ecx,qword ptr ds:[rax+70]	
0000000076F72C81	48 89	mov qword ptr ss:[rsp+20],rdx	
0000000076F72C86	8D 50	lea edx,qword ptr ds:[rax-1]	
0000000076F72C89	41 83	or r9d,FFFFFFFF	
0000000076F72C8D	4C 8B	mov r8,rbx	
0000000076F72C90	E8 93	call <kernel32.CompareStringA>	
0000000076F72C95	83 F8	cmp eax,2	
0000000076F72C98	75 16	jne kernel32.76F72CB0	
0000000076F72C9A	48 8B	mov rcx,qword ptr ss:[rsp+60]	
0000000076F72C9F	48 8D	lea rdx,qword ptr ds:[76F5D700]	
0000000076F72CA6	B8 01	mov eax,1	
0000000076F72CAB	48 89	mov qword ptr ds:[rcx],rdx	
0000000076F72CAE	E8 02	jmp kernel32.76F72CB2	
0000000076F72CB0	33 C0	xor eax,eax	
0000000076F72CB2	48 83	add rsp,30	
0000000076F72CB6	5B	pop rbx	
0000000076F72CB7	C3	ret	

Figure 9 address:0000000076F72C77 getting system's physical memory

0000000076F73F9E	48 85	test rax,rax	
0000000076F73FA1	75 07	jne kernel32.76F73FAA	
0000000076F73FA3	89 3E	mov dword ptr ds:[rsi],edi	
0000000076F73FA5	E9 8D	jmp kernel32.76F74237	
0000000076F73FAA	48 8D	lea rdx,qword ptr ds:[76FC81A0]	0000000076FC81A0:"GetSecurityInfo"
0000000076F73FB1	48 8B	mov rcx,rax	
0000000076F73FB4	E8 DF	call <kernel32.GetProcAddress>	
0000000076F73FB9	48 8D	lea rdx,qword ptr ds:[76FC81B0]	0000000076FC81B0:"SetSecurityInfo"
0000000076F73FC0	49 8B	mov rcx,r13	
0000000076F73FC3	48 89	mov qword ptr ss:[rsp+78],rax	
0000000076F73FC8	E8 C8	call <kernel32.GetProcAddress>	
0000000076F73FCD	48 8D	lea rdx,qword ptr ds:[76FC81C0]	0000000076FC81C0:"GetSecurityDescriptorControl"
0000000076F73FD4	49 8B	mov rcx,r13	
0000000076F73FD7	48 89	mov qword ptr ss:[rsp+80],rax	
0000000076F73FDF	E8 94	call <kernel32.GetProcAddress>	
0000000076F73FE4	33 C9	xor ecx,ecx	
0000000076F73FE6	4C 8B	mov r13,rax	
0000000076F73FE9	48 8B	mov rax,qword ptr ss:[rsp+78]	
0000000076F73FEE	48 3B	cmp rax,rcx	
0000000076F73FF1	0F 84	je kernel32.76F74222	
0000000076F73FF7	4C 3B	cmp r13,rcx	

Figure 10 address:0000000076F73FAA getting system's security software information

000007FEF284EEA5	48 85	test rax,rax	
000007FEF284EEA8	74 15	je mscoree.7FEF284EEB	
000007FEF284EEAA	48 8D	lea rdx,qword ptr ds:[7FEF2B73FF8]	000007FEF2B73FF8:"GetUserDefaultUILanguage"
000007FEF284EEB1	48 8B	mov rcx,rax	
000007FEF284EEB4	FF 15	call qword ptr ds:[&getProcAddress]	
000007FEF284EEBA	48 85	test rax,rax	
000007FEF284EEBD	75 04	jne mscoree.7FEF284EEC3	
000007FEF284EEBF	48 83	or rax,FFFFFFFFFFFFFFFF	
000007FEF284EEC3	48 87	xchg qword ptr ds:[7FEF2B84F68],r	
000007FEF284EECA	48 8B	mov rax,qword ptr ds:[7FEF2B84F68]	
000007FEF284EED1	48 83	cmp rax,FFFFFFFFFFFFFFFF	
000007FEF284EED5	75 08	jne mscoree.7FEF284EEDF	
000007FEF284EED7	FF 15	call qword ptr ds:[&GetSystemDef	
000007FEF284EEDD	E8 02	jmp mscoree.7FEF284EEE1	
000007FEF284EEDF	FF D0	call rax	
000007FEF284EEE1	44 0F	movzx r11d,ax	
000007FEF284EEE5	B8 01	mov eax,1	
000007FEF284EEE8	44 89	mov dword ptr ds:[rbx],r11d	
000007FEF284EED	48 83	add rsp,20	
000007FEF284EEF1	5B	pop rbx	
000007FEF284EEF2	C3	ret	

Figure 11 address:000000007FEF284EEAA getting system's language information

It also reads the software policies to get the information of the software installed in the system.

```

0000000076F311A9 85 C0 test eax, eax
0000000076F311AB 78 8F js kernel32.7
0000000076F311AD 48 8D lea rdx, qword [0000000076FAF980:L"\\Software\\Policies\\Microsoft\\Windows\\Safer\\CodeIdentifiers"]
0000000076F311B4 48 8D lea rcx, qword
0000000076F311B5 FF 13 lea rax, qword ptr
0000000076F311B8 85 C0 test eax, eax
0000000076F311C1 78 59 js kernel32.7
0000000076F311C3 48 8D lea rax, qword
0000000076F311C8 4C 8D lea r8, qword [rsp+8]: "PE"
0000000076F311D0 48 8D lea rcx, qword
0000000076F311D5 41 8B mov ebx, r12d
0000000076F311D8 C7 84 mov dword ptr [rsp+8]: "PE", 30: '0'
0000000076F311E3 4C 89 mov qword ptr [rsp+B8]: L"C:\\Users\\worker\\Desktop\\"
0000000076F311EB 48 89 mov qword ptr [rsp+B8]: L"C:\\Users\\worker\\Desktop\\"
0000000076F311F3 C7 84 mov dword ptr 40: 'e'
0000000076F311FE 4C 89 mov qword ptr
0000000076F31206 4C 89 mov qword ptr
0000000076F3120E FF 15 call qword ptr
0000000076F31214 85 C0 test eax, eax
0000000076F31216 0F 89 js kernel32.
0000000076F3121C 65 48 mov rax, qword
0000000076F31225 4C 8B mov r8, qword

```

Figure 12 address:0000000076F311AD reading software's policies

C&C Server

The RAT connects to the C2 server and sends the collected data from the victim's system to the C2 server, whose IP addresses are:

192.3.122.73

45.61.48.65

The server port numbers used are "1339" and "6767".

The malware keeps trying to make connections to the two IP addresses until the connection has been established.

No.	Time	Source	Destination	Protocol	Length	Info
3322	5991.560866	10.0.2.15	45.61.48.65	TCP	385	49622 → 6767 [PSH, ACK] Seq=1 Ack=2 Win=65535 Len=331
3323	5991.561134	45.61.48.65	10.0.2.15	TCP	60	6767 → 49622 [ACK] Seq=2 Ack=332 Win=65535 Len=0
3324	5993.908030	10.0.2.15	45.61.48.65	TCP	54	49622 → 6767 [FIN, ACK] Seq=332 Ack=2 Win=65535 Len=0
3325	5993.908375	45.61.48.65	10.0.2.15	TCP	60	6767 → 49622 [ACK] Seq=2 Ack=333 Win=65535 Len=0
3326	5993.910311	10.0.2.15	192.3.122.73	TCP	66	49623 → 1339 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=16 SACK_PERM=1
3327	5996.914260	10.0.2.15	192.3.122.73	TCP	66	[TCP Retransmission] 49623 → 1339 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=16 SACK_PERM=1
3328	6002.913234	10.0.2.15	192.3.122.73	TCP	62	[TCP Retransmission] 49623 → 1339 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1
3329	6017.411642	10.0.2.15	45.61.48.65	TCP	66	49624 → 6767 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=16 SACK_PERM=1
3330	6017.728352	45.61.48.65	10.0.2.15	TCP	60	6767 → 49624 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460
3331	6017.728494	10.0.2.15	45.61.48.65	TCP	54	49624 → 6767 [ACK] Seq=1 Ack=1 Win=65535 Len=0
3332	6018.040723	45.61.48.65	10.0.2.15	TCP	60	6767 → 49624 [FIN, ACK] Seq=1 Ack=1 Win=65535 Len=0
3333	6018.040789	10.0.2.15	45.61.48.65	TCP	54	49624 → 6767 [ACK] Seq=1 Ack=2 Win=65535 Len=0
3334	6018.814634	10.0.2.15	45.61.48.65	TCP	385	49624 → 6767 [PSH, ACK] Seq=1 Ack=2 Win=65535 Len=331
3335	6018.814859	45.61.48.65	10.0.2.15	TCP	60	6767 → 49624 [ACK] Seq=2 Ack=332 Win=65535 Len=0
3336	6021.165670	10.0.2.15	45.61.48.65	TCP	54	49624 → 6767 [FIN, ACK] Seq=332 Ack=2 Win=65535 Len=0
3337	6021.166173	45.61.48.65	10.0.2.15	TCP	60	6767 → 49624 [ACK] Seq=2 Ack=333 Win=65535 Len=0
3338	6021.167986	10.0.2.15	192.3.122.73	TCP	66	49625 → 1339 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=16 SACK_PERM=1

Frame 3323: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0
 Ethernet II, Src: RealtekU_12:35:02 (52:54:00:12:35:02), Dst: PcsCompu_5e:c5:62 (08:00:27:5e:c5:62)
 Internet Protocol Version 4, Src: 45.61.48.65, Dst: 10.0.2.15
 Transmission Control Protocol, Src Port: 6767, Dst Port: 49622, Seq: 2, Ack: 332, Len: 0

```

0000 08 00 27 5e c5 62 52 54 08 12 35 02 08 00 45 00  ..^.BRT...E.
0010 00 28 03 af 00 00 40 06 00 95 2d 3d 30 41 0a 00  ..(..@...--0a.
0020 02 0f 1a 6f c1 d6 2b cd de 03 5f bf 99 24 50 10  ...o.+...$P.
0030 ff ff 67 4d 00 00 00 00 00 00 00 00 00 00 00 00  ..gM.....

```

Figure 13 connecting to C2 server

Once the connection to the C&C server is established, the information that is collected by the malware is sent to its server. The packet that is sent from the victim's machine to the server looks like the image below:

Most of the strings in the packet are base64 encoded.

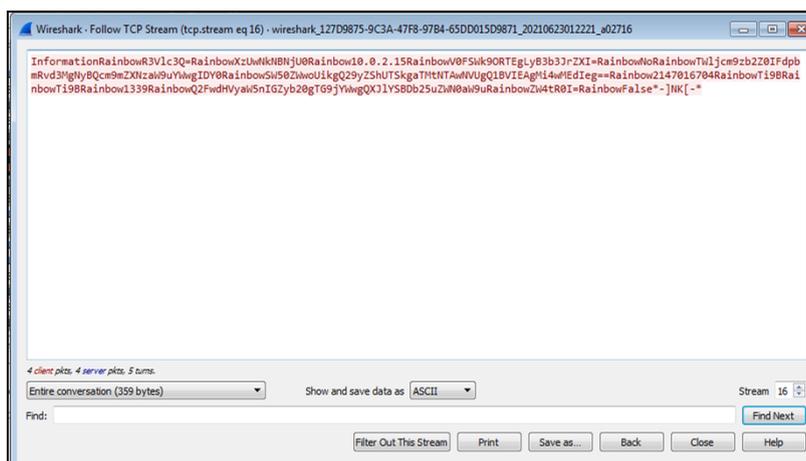


Figure 14 packet sent to server

InformationRainbowR3Vlc3Q=RainbowXzUwNkNBNjU0Rainbow10.0.2.15RainbowV0FSWk9
 ORTEgLyB3b3JrZXI=RainbowNoRainbowTWljcm9zb2Z0IFdpbmRvd3MgNyBQcm9mZXNzaW
 9uYWwgIDY0RainbowSW50ZWwoUikgQ29yZShUTSkgatMtNTAwNVUgQ1BVIEAgMi4wMEDl
 eg==Rainbow2147016704RainbowTi9BRainbowTi9BRainbow1339RainbowQ2FwdHVyaW5n
 IGZyb20gTG9jYWwgQXJlYSBDb25uZWNoaW9uRainbowZW4tR0I=RainbowFalse*-]NK[-*

As we can see, the packet is split into different blocks which are separated by the string “Rainbow”.

Below is the explanation of different blocks present in the packet:

- ⑩ “Rainbow” is a separator here which splits the data in the packet.
- ⑩ “Information” is always the first part of the packet.
- ⑩ “R3Vlc3Q=” is decoded as “Guest”.
- ⑩ “XzUwNkNBNjU0” is decoded as “_506CA654”, which is the volume information.
- ⑩ “10.0.2.15” is the IP address of victim’s machine.
- ⑩ “V0FSWk9ORTEgLyB3b3JrZXI=” is decoded as “WARZONE1 / worker” is the victim’s machine name nad username.
- ⑩ “No” is whether the victim has a webcam or not.
- ⑩ “TWljcm9zb2Z0IFdpbmRvd3MgNyBQcm9mZXNzaW9uYWwgIDY0” is decoded as “Microsoft Windows 7 Professional 64”, which is the victim’s Windows system information.
- ⑩ “SW50ZWwoUikgQ29yZShUTSkgatMtNTAwNVUgQ1BVIEAgMi4wMEDleg==” is decoded as “Intel(R) Core(TM) i3-5005U CPU @ 2.00GHz”, which is the CPU information.
- ⑩ “2147016704” is the system physical memory.
- ⑩ “Ti9B” is decoded as “N/A”, which could be whether any antivirus or firewall is installed or not.
- ⑩ “1339” is the port number of the C&C server it is connecting to.
- ⑩ “Q2FwdHVyaW5nIGZyb20gTG9jYWwgQXJlYSBDb25uZWNoaW9u” is decoded as “Capturing from Local Area Connection” which could be the title of the top-most window, in mine it was Wireshark.
- ⑩ “ZW4tR0I=” is decoded as “en-GB”, which is the language used on the victim’s machine.
- ⑩ “*.-]NK[-*” is used to define the end of packet.

Sandbox Evasion

It tries to detect if any VM or sandbox is running in the system to evade the virtual environment. It also drops some PE files which has not been started or loaded, to remove the attention from the actual backdoor.

```
-----  
Files added:11167  
-----  
C:\cmdex.exe  
C:\Documents and Settings.exe  
C:\escw_103_sa.exe  
C:\MSOCache.exe  
C:\PDFStreamDumper.exe  
C:\PerfLogs.exe  
C:\Recovery.exe  
C:\temp.exe
```

Figure 15 dropped PE files

MITRE Attack Techniques Used

Technique ID	Tactic	Technique
T1106	Execution	Native API
T1037.005	Persistence	Startup Items
T1547.001	Persistence	Registry Run Keys / Startup Folder
T1055	Privilege Escalation	Process Injection
T1036	Defensive Evasion	Masquerading
T1497	Defensive Evasion	Virtualization/Sandbox Evasion
T1564.001	Defensive Evasion	Hidden Files and Directories
T1027	Defensive Evasion	Obfuscated Files or Information
T1027.002	Defensive Evasion	Software Packing
T1518.001	Discovery	Security Software Discovery
T1083	Discovery	File and Directory Discovery
T1082	Discovery	System Information Discovery
T1560	Collection	Archive Collected Data
T1573	Command and Control	Encrypted Channel
T1571	Command and Control	Non-Standard Port
T1219	Command and Control	Remote Access Software

Sectrio Protection

Sectrio detects this malware as “SS_Gen_RevengeRAT_PE_A”.

IOC's

IPs
192.3.122.73
45.61.48.65

Sample
313afaedd435018403f358337115eb45

Dropped file
f3dd3b35b5ea9f30f946f86e6b3a4730
c2082df303350b29360581d4bf3e9d42
3199e7449063b2699b48cdf607357bfc
7fa50220d182c6ec9e223355e8daf2bd
3e2bfc63f1945a9c2bd43822ea609cda
3cab7117fcc6f319e2ccfc550e8b284a

Our Honeypot Network

This report has been prepared from the threat intelligence gathered by our honeypot network. This honeypot network is today operational in 72 cities across the world. These cities have at least one of the following attributes:

- Are landing centers for submarine cables
- Are internet traffic hotspots
- House multiple IoT projects with a high number of connected endpoints
- House multiple connected critical infrastructure projects
- Have academic and research centers focusing on IoT
- Have the potential to host multiple IoT projects across domains in the future

Over 12 million attacks a day is being registered across this network of individual honeypots. These attacks are studied, analyzed, categorized, and marked according to a threat rank index, a priority assessment framework that we have developed within Sectrio. The honeypot network includes over 4000 physical and virtual devices covering over 400 device architectures and varied connectivity mediums globally. These devices are grouped based on the sectors they belong to for purposes of understanding sectoral attacks. Thus, a layered flow of threat intelligence is made possible.