Malware Analysis – PlugX – Part 2

countuponsecurity.com/2018/05/09/malware-analysis-plugx-part-2/

By Luis Rocha

Following my previous article on PlugX, I would like to continue the analysis but now use the PlugX controller to mimic some of the steps that might be executed by an attacker. As you know the traditional steps of an <u>attack lifecycle</u> follow, normally, a predictable sequence of events i.e., Reconnaissance, initial compromise, establish foothold, escalate privileges, internal reconnaissance, move laterally, maintain persistence, complete mission. For sake of brevity I will skip most of the steps and will focus on the lateral movement.I will use the PlugX controller and C2 functionality to simulate an



attacker that established a foothold inside an environment and obtained admin access to a workstation. Following that, the attacker moved laterally to a Windows Domain Controller. I will use the PlugX controller to accomplish this scenario and observe how an attacker would operate within a compromised environment.

As we saw previously, the PlugX controller interface allows an operator to build payloads, set campaigns and define the preferred method for the compromised hosts to check-in and communicate with the controller. In the PlugX controller, English version from Q3 2013, an operator can build the payload using two techniques. One is using the "DNS Online" technique which allows the operator to define the C2 address e.g, an URL or IP address, that will be used by the payload to speak with the C2. The other method, is the "Web Online", which allows the operator to tell the payload from where it should fetch the C2 address. This method allows the operator to have more control over the campaign. The following diagram illustrates how the "Web Online" technique works.

May 9, 2018



Why do I say this technique would allow an attacker to have more control? Consider the case that an organization was compromised by a threat actor that use this PlugX technique. In case the C2 is discovered, the impacted organization could block the IP or URL on the existing boundary security controls as a normal reaction to the concerns of having an attacker inside the network. However, the attacker could just change the C2 string and point it to a different system. In case the organization was not able to scope the incident and understand the TTP's (Tools, Tactics and Procedures) then the attacker would still maintain persistence in the environment. This is an example that when conducting incident response, among other things, you need to have visibility into the tools and techniques the attacker is using so you could properly scope the incident and perform effective and efficient containment and eradication steps. As an example of this technique, below is a print screen from a GitHub page that has been used by an unknown threat actor to leverage this method.

()	Featur	es Business	Explore	Marketplace	Pricing	This repository					Sign in or S	iign up
Ģ		/ news					0	Watch	0	🖈 Star	0 ¥ Fo	rk 0
↔ Co	de	1 Pull requests 0	l <u>dt</u> Ins	ghts								
Upda P mast	te er	htm									Browse	files
₩.	_	committed on Ja	an 8 Verified			1 parent 69a21	bd commit		_	_	cfbd	568a84
Show	ving 1 d	nanged file with 1	addition an	d 1 deletion.							Unified	Split
2		.htm				Attacker modified the C2 string for					View	~
		(0 -1,5 +1,5 (0)		/	PlugX o	n a GitH	ub pag	ge			
1	1	<html></html>										
2		-cl DZKSjN		oendm=dz35>	-							
	2	+ DZKSjd</td <th></th> <th>IGNDQzDZJS></th> <td></td> <td></td> <th></th> <th></th> <th></th> <th></th> <th></th> <td></td>		IGNDQzDZJS>								
3	3	***********	*****	************	***********		**********	******				
- 4	4	Welcome to	GitHub Pag	es(br>								
5	5	You can use th	e [editor o	n GitHub]								
2	(z											
0 com	nents o	on commit 44b	a2d3									

Please sign in to comment.

So, how to leverage this technique on the PlugX builder? The picture below shows how the operator could create a payload that uses the "Web Online" technique. The C2 address would be fetched from a specified site e.g. a Pastebin address, which on its turn would redirect the payload to the real C2 address. The string

"DZKSAFAAHHHHHHOCCGFHJGMGEGFGCHOCDGPGNGDZJS" in this case is used to decode to the real C2 address which is "www.builder.com". On the "<u>PlugX: some uncovered</u> <u>points</u>" article, Fabien Perigaud writes about how to decode this string. Palo Alto Unit42 gives another example of this technique on the "<u>Paranoid PlugX</u>" article. The article "<u>Winnti</u> <u>Abuses GitHub for C&C Communications</u>" from <u>Cedric Pernet</u> ilustrates an APT group leveraging this technique using GitHub.

🗢 LZ-1(2013-8-18)				
Users Listener Builder Setting	About			
First DNS Online Web Online	Prepare DNS Prepare Proxy	Install Inject Schedule	ScreenRecorder Join Op	tion
ummer (a pastebin.com/YxA	FXsp1			
HTTP//: P				
HTTP//:				
HTTP//: I				_
HTTP//:				
Host- www.builder.com	Doxt:	80 E TCP		
	Port			1.54
DZKSAFAAHHH	HOCCGFHJGMGEGFGGHOCDG	PGNGDZ3S		
EXE	Binary	C Array	PAS Array	
Int. Ithe Connect				
Time	Computer	Event	Daramitor	
1 € 2018-04-10 20:05:58	*	Listen start success!	Palatike	
2018-04-10 20:06:10	•	Listen stop success!		
€ 2018-04-10 20:06:21	•	Listen start successi		
7				
	and the second	Carbon C	010	
2018-04-10 20:07:47 Listenne	r: 0 Computer: 1	Online: 0	Offine: 1	10

For sake of simplicity, in this article, I'm going to use the DNS Online technique using "www.builder.com" as C2 address. Next, on the "First" tab I specify the campaing ID and the password used by the payload to connect to the C2.

TIMEOUT!!!								
Users Listener Builder Sett	ing About							
First DNS Online Web Onli	ine Prepare DNS	Prepare Proxy Ins	tall Inject Schedule	ScreenRecorder	Join Option			
Online Pass: random	password							
018101000.1								
Online Memo: Operati	ionUSA							
Online Mutex: Plugx		•						
EXE		Binary	C Array		PAS Array			
						- 1		
In In In In								
LOG HUX Connect								
Time Abacia of to control to	Computi	Br E	event		Paramiter	_^		
9/2018-04-10 22:50:18	CH2HDC0		omputer online					
42018-04-10 22:50:18	042423		mputer office					
2018-04-13 10:18:18	01671000	99 0	mputer offine					
						×		
¢		11				2		
2018-04-13 13:54:22 Lis	tenner: 1	Computer: 2	Online: 0	Offline: 2		1		

Next, on the Install tab I specify the persistence settings, in this case I'm telling the payload to install a service and I can specify different settings including where to deploy the binaries, the service name and service description. In addition, I can specify that if the Service persistence mechanism fails due to some reason the payload should install the persistence mechanism using the Registry and I can specify which HIVE should be used.

🗢 LZ-1(2013-8-18)								
Users Listener Builder	Setting About							
First DNS Online Web	First DNS Online Web Online Prepare DNS Prepare Proxy Install Inject Schedule ScreenRecorder Join Option							
	Service Or Register	C Service (Only C Register C	nly C None				
Install Dir:	%AUTO%\RasTIs				•			
	,				_			
Server Name:	RasTis							
Server Disp:	RasTls							
Server Desc:	Symantae 802 1v Sympleant							
	,-,							
Key Root:	HKEY_CURRENT_USER				•			
EVE	1	Binary	C Array	PA	S Array			
		Denary			(5 Alla)			
ter brite it								
Log Flux Connect								
Time	Computi	ar E\	vent	Pa	aramiter			
4 2018-04-10 20:05:58	:	List	ten start success!					
12018-04-10 20:06:10		List	ten stop successi					
Contraction and Contract		25						
<						>		
2018-04-10 20:09:17	Listenner: 0	Computer: 1	Online: 0	Offline: 1		1		

Then, In the inject tab I specify which process will be injected with the malicious payload. In this case I choose "svchost.exe". This will make PlugX start a new instance of "svchost.exe" and then inject the malicious code into svchost.exe process address space using <u>process</u> hollowing technique.

🗢 LZ-1(2013-8-18)				
Users Listener Builder	Setting About			
First DNS Online Web	Online Prepare DNS Prepare Pro	xy Install Inject Schedule	ScreenRecorder Join Option	
	 Inject Nor 	18		
App1:	%windir%\system32\svchost.exe			
App2:)			-	
App3:			<u>.</u>	
App4:			•	
EXE	Binary	C Array	PAS Array	1
Log Flux Connect				
Time	Computer	Event	Paramiter	
1 2018-04-10 20:05:58	•	Listen start success!		
12018-04-10 20:06:10		Listen stop success!		
2018-04-10 20:06:21	•	Listen start success!		
<		11		×
2018-04-10 20:11:51	Listenner: 0 Computer: 3	1 Online: 0	Offline: 1	11.

Other than that, the operator could define a schedule and determine which time of the week the payload should communicate with the C2. Also the operator could define the Screen Recording capability that will take screenshots at a specific frequency and will save them encrypted in a specific folder.

🗢 LZ-1 (2013-8-18)			
Users Listener Builder Setting	About		
First DNS Online Web Online	Prepare DNS Prepare Proxy	Install Inject Schedule ScreenRec	order Join Option
			[
	Enable		
Fre	equency: 10 seconds		
	Zoom: 50		
c	Color bits: 16 💌		
	Quality: 50		
Rem	nain days: 3		
	Save Dir: %AUTO%\screen		
	1		
EXE	Binary	C Array	PAS Array
Log Flux Connect			
Time	Computer	Event	Paramiter
X 2018-04-19 10:07:51	*	Listener protocol start fail[0]The ope	eration
€ 2018-04-19 10:07:51	•	Listen start success!	
	Ш		>
2018-04-19 10:08:01 Listen	nner: 1 Computer: 2	Online: 0 Offline: 2	1

Last settings on the "option" tab allow the operator to enable the keylogger functionality and specify if the payload should hide it self and also delete itself after execution.

🗢 LZ-1(2013-8-18)										
Users Listener Buil	der Setting Ab	out								
First DNS Online	Web Online Pre	apare DNS	Prepare Proxy	Instal	Inject	Schedule	ScreenRecor	der Join	Option	
Keylogger										
Hide Service	:e									
🔽 Delete Sel	f									
				1			1			_
	EXE		Binary			C Array		PAS	Array	
Log Flux Conne	ect									
Time		Compute	ar	Eve	nt			Para	miter	
\$ 2018-04-19 10:07	:51	8		Lister	ner proto	col start fai	II[0]The opera	ation		
02018-04-19 10:07	:51	*		Lister	n start su	ccess!				
			ш	_						>
2018-04-19 10:14:22	Listenner:	1	Computer: 2		Online: 0	1	Offline: 2			

Finally, after all the settings defined, the operator can create/download the payload in different formats. An executable, binary form (Shellcode), or an array in C that can then be plugged in another delivery mechanism e.g, PowerShell or MsBuild. After deploying and installing the payload on a system, that system will check-in into the PlugX controller and an operator can call the "Manager" to perform the different actions. In this example I show how an attacker, after having compromised a system, uses the C2 interface to:

Browse the network

• [۵	🗢 [CHEZH2399]Manager 📰 🖸 🔀												
Disk	Nethood FileTran	sfer Process	Services	Register	Netstat	Capture	Control	Shell	ShellEmluator	PortMap	SQL	KeyLogger	Option
	Network			WorkGroup			Computers			Share			
	() ()			2		3				2			
VM	ware Shared Folders		COUN	π		OHEZH	2399		NETLOG	N			
	()					8							
Micro	osoft Terminal Service	5				снано	C01		SYSVO	L			
Micro	soft Windows Netwo	ſk											
,	(Neb Client Network)												
folthe	operation completed su	restully.											

Access remote systems via UNC path

🗢 [CHEZH2399]Manager				
Disk Nethood FileTransfer Process Se	arvices Register Netstat	Capture Controll Shell	ShellEmluator PortMap SQ	L KeyLogger Option
				Go 2 1 1 1 1
B- S CHEZH2399 B- S (C:)Hard disk				
🗄 🎱 (D:)CDROM(DVD)	\$Recycle.Bin Boot	Documents PerfLogs and Settings	Program Files ProgramData	System Users Volume I
		2	2	
	Windows autoexec	bootmgr BOOTSEC	. config.sys pagefile.sys	
[0]The operation completed successfully.				

Upload and execute a file e.g., upload PlugX binary

🗢 [CHE2H2399]Manager				
Disk Nethood FileTransfer Process Services Reg	kter Netstat Capture Controll Shell ShellEmiu	ator PortMap SQL KeyLogge	r Option	
Start Stop Delete	Start All Stop All Delete All			
Remote Path		Progress	Speed	Transfered
Ship to the second seco		100.00%		265.39 KB
<				2
[0]The operation completed successfully.				

Invoke a command shell and perform remote commands e.g., execute PlugX binary on a remote system



Previous pictures illustrate actions that the attacker could perform to move laterally and, for example, at some point in time, access a domain controller via UNC path, upload the PlugX payload to a directory of its choice and execute it. In this case the pictures show that the PlugX payload was dropped into c:\PerfLogs\Admin folder and then was executed using WMI. Below example shows the view from the attacker with two C2 sessions. One for one workstation and another for a domain controller.

						🖸 🛎 (0630399 Messager 🖉 🗖 🗖
Users Lictorier Builder	Setting About					Disk Nerhood NeTranske Process Services Register Netztat Capture Control Shell ShellCritiator PortHap 9QL (1).4
Computer I+	Memo	Lan	Han	Location		But III C Structure & Structure
E Default00/03						
					_	Process 'CE Pol Protocol L-IP L-Port State R-IP
8 OperationUSA	2/40					≪ System Ide Process 0 TOP 10.0.0.20 50071 TME-WAIT 172.16.0.300
-					_	System 4 TOP 0.0.00 Attacker broasing the network D.0.0
062639	OperationUSA	10.0.0.20	10.0.0.20		_	Sectors 4 TOP 0.0.00 consections on a workstation 0.0.0.0
100					_	- Settern 4 TOP 1546-20 0.0.0.0
30 000 00	OperationUSA	33-0-0-200	20.0.0.100			0 Sector 4 (CP 1100.20 137 · · ·
					_	w system w GDP 2000.20 Link
					_	
						Nilliae sheraron contraint accessingly.
						- (OUNCOL) Manager
						Dick Nethood ReTransfer Process Services Register Netstal Capture Control 1948 Shellimiuator PortHap 903. 11
					_	Microsoft Windows (Winston 6.0 6000)
						Copyright (c) 2006 Microsoft Corporation. All rights reserved.
c	-				2	C/Windowijpynen205.cd \
Log Rux Connect						Chupcordg Jal Attacker involving a shall on a
Time	Computer	Event		Pauritor	~	Domain Centroller using the C2
4 2018-04-10 22:50:00	1	Laten stop success!			_	interface
Q #2018-04-10 22:50:10		Listen start success?				
042018-04-1022:50.18	04240001	Computer online				
0 2010-04 10 22:50:10	04292309	controlet onere			×	v l
(C						
2018-04-10 22:51:12	Listermer: 1 Computer	2 Online: 2	Offine: 0		1	DCDra operation completed auccessfully.

Having access to a domain controller is likely one of the goals of the attacker so he can obtain all the information he needs about an organization from the Active Directory database.



To access the Active Directory database, the attacker could, for example, run the "ntdsutil.exe" command to create a copy of the "NTDS.dit" file using Volume Shadow Copy technique. Then, the attacker can access the database and download it to a system under his control using the PlugX controller interface. The picture below illustrates an attacker obtained the relevant data that was produced using the "ntdsutil.exe" command.



Finally, the attacker might delete the artifacts that were left behind on the file system as consequence of running "ntdsutil.exe".



So, in summary, we briefly looked at the different techniques a PlugX payload could be configured to speak with a Command and Controller. We built, deploy and install a payload. Compromised a system and obtain a perspective from PlugX operator. We move laterally to a domain controller and installed the PlugX payload and then used a command shell to obtain the Active Directory database. Of course, as you noted, the scenario was accomplished with an old version of the PlugX controller. Newer versions likely have many new features and capabilities. For example, the print screen below is from a PlugX builder

from 2014 (MD5: 534d28ad55831c04f4a7a8ace6dd76c3) which can create different payloads that perform <u>DLL Search order hijacking</u> using Lenovo's RGB LCD Display Utility for ThinkPad (tplcdclr.exe) or Steve Gibson's Domain Name System Benchmarking Utility (sep_NE.exe). The article from Kaspersky "<u>PlugX malware: A good hacker is an apologetic hacker</u>" outlines a summary about it.

🔒 FAST统一生成器	v2014.2.24	×
保存配置数据 表 名称 域名上线	収配置数据 ======EXE_SC====== =====EXE====== [备用DNS 时间表 其他逃项	
安装目	%ALLUSERSPROFILE%\Intel(R) Capability Licensing Service Interfac	TPLCDCLR_v4.0(停用) DNSBENCH_v2.2(WIN2K专用)
服务名:	Intel(R) Capability Licensing Service Interface CPUMonitor	ADOBE V2.3.1 CreateShort v2.3.2 cen NE v3.1.2
显示名:	Intel(R) Capability Licensing Service Interface CPUMonitor	DNSBENCH_v31.3
请确保	安装目录的父目录存在且可写的。否则FAST会卡死	This PlugX builder from 2014 gives the operator the ability to create 2 different "Trinity" payloads
HEAD安装	義后的EXE伪装名 MediaCenter.exe	

That's it! With this article we set the table for the next article focusing on artifacts that might helps us uncover the hidden traits that were left behind by the attacker actions performed during this scenario. Stay tuned and have fun!