The Italian Connection:
An analysis of exploit supply chains and digital quartermasters

by
Ned Moran (ned@shadowserver.org)
Ben Koehl (benk@accessviolation.org)

Overview

On July 5, 2015 an unknown hacker publicly announced on Twitter that he had breached the internal network of Hacking Team - an Italian pentesting company known to purchase 0-day exploits and produce their own trojans. The hacker proceeded to leak archives of internal Hacking Team tools and communications. A number of tools and previously unknown exploits were discovered in the trove of data posted online.

In this paper we will focus on two exploits which at the time of discovery in the Hacking Team archives were unpatched. The two 0-days in question targeted Adobe Flash and were subsequently labeled CVE-2015-5119[1] and CVE-2015-5122[2].

The goal of this research is to demonstrate how quickly these exploits spread and were used by multiple independent cyber espionage operators.[3] Via the evidence presented within this paper we will demonstrate that at least two different exploit kits, or generators, were constructed by an unknown entity and shared amongst multiple operators believed to be located in China. We believe the following is a clear example of yet another 'digital quartermaster' of cyber espionage tools.

Research Methodology

For this research we set out to collect as many CVE-2015-5119 and CVE-2015-5122 exploits as possible. We excluded exploits that were delivered by popular crimeware kits such as Angler. We chose to focus our efforts on exploits used in a more targeted fashion by cyber espionage operators.

Our approach to data collection was two-fold. First, we crawled specific websites that have been previously used to deliver exploits and malware in 'strategic web compromise (SWC)' or 'watering hole' attacks[4]. Second, we deployed a variety of Yara signatures designed to detect malicious Flash files that exploited both CVE-2015-5119 and CVE-2015-5122 into repositories like VirusTotal[5] and Shadowserver. We collected a total of 52 unique samples via these techniques.

Once collected, we set about designing a process to cluster our data set. For each file collected we enumerated the following data points where possible:

- SWF MD5: MD5 hash of the Flash exploit file
- CVE: Common Vulnerabilities and Exposures identifier

---

[1] https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2015-5119

[2] https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2015-5122

[3] We define independent operators as actors that maintain distinct infrastructure without any technical overlaps such as ip history.

[4] An example of a previous Strategic Web Compromise campaign can be found at http://blog.shadowserver.org/2012/05/15/cyber-espionage-strategic-web-compromises-trusted-websites-serving-dangerous-results/

[5] http://www.virustotal.com/intelligence/

- Last-modified data: if we discovered the exploit in the wild we collected the last-modified data of the file as a means to determine when the attacker released the exploit into the wild
- Referrer site (aka SWC or Watering hole): where possible we noted the referrer url
- Exploit site: where possible we noted the site where the exploit was hosted
- Creation date: if metadata was present we noted the creation date of the Flash file
- ActionScript class name: classname of ActionScript that invokes the exploit
- ActionScript hashes: the md5 hash of each ActionScript class embedded in the Flash file
- Compression: the compression algorithm used on the Flash file
- Payload location: the location of the payload (e.g. in the Flash file or on a remote server)
- Payload MD5: MD5 hash
- C2 server: the command and control server for the dropped payload
- Attribution: where possible we noted the actor responsible for deploying the exploit into the wild

In this paper we are less interested in a detailed analysis of individual actors using these exploits. Rather our goal is to reveal relationships between groups that appear to be independent based on analysis of standard technical artifacts such as dropped payload and command and control infrastructure. Where we could not definitively attribute an exploit to a specific actor, we were comfortable in simply asserting that the artifacts from the exploit to payload chain in question did or did not overlap with other exploit to payload chains in our collection.

As we clustered the Flash files in our data set we identified five distinct sets of exploits. We believe that two of these clusters of files were created by two different privately shared exploit kits or generators tools. These kits or generators are a graphical user interface or command line tool. The tool enables an operator to quickly and easy bind a payload or remote download url to shellcode in the flash exploit file via a handful of mouse clicks or a simple command.

Our data illustrates that distinct intrusion sets or actors were using the same exploit variants from the same kits on the same day in different attacks leveraging different infrastructure. This finding suggests that multiple kits or generators were shared with independent actors.

Throughout the body of this paper we will describe each cluster of exploit activity that we observed. We will then conclude with an overall analysis of this activity and present several competing hypotheses that can be used to explain the exploit clusters that we observed.

Cluster Analysis

The five clusters of exploit activity that we discovered will henceforth be referred to as HT_Exploit cluster, flash_exploit_002 cluster, exp1_fla cluster, exp2_fla cluster and finally movie_fla cluster. Each of these clusters were named for the ActionScript class used to invoke the exploit code.

Cluster 1: HT_Exploit

Shortly after the July 5, 2015 public announcement of the intrusion into the Hacking Team's network and subsequent release of the company's data, the first 0-day exploit targeting Adobe Flash was discovered. On July 7, 2015 Adobe released a patch for the underlying vulnerability targeted by the discovered exploit code.[6] This vulnerability was labeled CVE-2015-5119. Shortly after this patch was released a number of different cyber espionage operators deployed exploits targeting this vulnerability.

In total we collected 13 Flash files that exploited CVE-2015-5119 and according to the file's metadata were created on July 7, 2015. Of these 13 files, as shown in Table 1, 11 files had the same ActionScript class name of HT_Exploit. These 11 files were all compressed with the LZMA algorithm and had the malicious payload bundled internally.

Date Created	SWF MD5	AS Class	CVE	Compression	Embedded Payload
7/7/2015	dceae0d1a680bc098bae9da466e12610	HT_Exploit	CVE-2015-5119	LZMA	yes
7/7/2015	5392f1399a49935817669d22e5e644ea	HT_Exploit	CVE-2015-5119	LZMA	yes
7/7/2015	da6c98d8f37290a10119fbca33eec58a	HT_Exploit	CVE-2015-5119	LZMA	yes
7/7/2015	878d13b8ceb49cfe9ff1b063bffeb9a9	HT_Exploit	CVE-2015-5119	LZMA	yes
7/7/2015	079a440bee0f86d8a59ebc5c4b523a07	HT_Exploit	CVE-2015-5119	LZMA	yes
7/7/2015	2c6126e9f308d1be11553978e8a97621	HT_Exploit	CVE-2015-5119	LZMA	yes
7/7/2015	75dc1e22e16c39e3532673f75fd41b93	HT_Exploit	CVE-2015-5119	LZMA	yes
7/7/2015	00591821f328911380277272164d08cd	HT_Exploit	CVE-2015-5119	LZMA	yes
7/7/2015	0b3a047d31461e20887bb1d32b4e472f	HT_Exploit	CVE-2015-5119	LZMA	yes
7/7/2015	f46019f795bd721262dc69988d7e53bc	HT_Exploit	CVE-2015-5119	LZMA	yes
7/7/2015	79dc5ee17ab11a647d6dff51d3908bda	HT_Exploit	CVE-2015-5119	LZMA	yes

Table 1: HT_Exploit Cluster

Further, each of these 11 Flash files had identical ActionScript classes as shown in Table 2 below:

[6] https://helpx.adobe.com/security/products/flash-player/apsa15-03.html

ActionScript classname	MD5
§bin_bin$cdc90048eba972f1f617b202a379b8d8-1052822192§.as	b5847d4f60ecba8a09a019d8826a6a18
HT_exploit.as	55bc2ac6bfcaaf9364a67cbd837aa66e
MyClass.as	3652a267b318b13c99c1a817416406ee
MyClass1.as	4b705980ed1b07becd76f47e007b5b3a[7]
MyClass2.as	955de95974a6228846cea327772815fe
MyUtils.as	23489ab7e77f7c69db3e2c6fd791bddb
ShellWin32.as	2d34c498fa0a65a59fd724d1d5466fbc

Table 2: HT_Exploit ActionScript classes

In summary the 11 Flash exploit files listed in Table 1 share the following characteristics:

- Created on the same date of 7/7/2015
- Targeted the same vulnerability of CVE-2015-5119
- Compressed via the LZMA algorithm
- Contained an embedded payload
- Had identical ActionScript as shown in Table 2

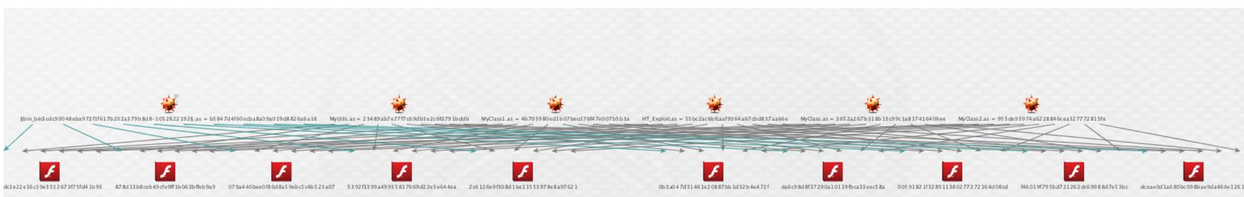

Image 1: HT_Exploit Cluster

The common attributes strongly suggests these Flash exploits were created by a single exploit generator. Further analysis of the payloads dropped by these exploits suggests that this single exploit generator was privately shared amongst a number of different intrusion operators. Although attribution was not our focus, we were able to conclusively attribute a number of malicious Flash files to different known cyber espionage operators. Where we were unsure regarding attribution we simply labeled the exploit to payload chain as 'unknown' followed by a number to distinguish between different sets of unknown activity.

[7] This ActionScript class was seen in the metasploit module targeting CVE-2015-5119

Actor	SWF MD5	Payload MD5	Payload Family	C2 Server
unknown 1	79dc5ee17ab11a647d6dff51d3908bda	af0d365a2c59709ece196037740bdb81	T5000	www.mcafeea.cf
wekby/APT18	079a440bee0f86d8a59ebc5c4b523a07	cfbcb83f8515bd169afd0b22488b4430	gh0st	223.25.233.248
menupass/APT10	da6c98d8f37290a10119fbca33eec58a	f8b3ad7d73ba432bc3e7084f9f7dee7d	Unknown	sbuudd.webssl9.info
unknown 2	f46019f795bd721262dc69988d7e53bc	b3bc4b5f17fd5f87ec3714c6587f6906	emdivi	www.n-fit-sub.com
unknown 3	2c6126e9f308d1be11553978e8a97621	0d50bd8299de64525a78845957456959	HTTPBrowser	dns.snakesearch.info
unknown 4	75dc1e22e16c39e3532673f75fd41b93	6739542294a6cc5ca4f272181944b943	HTTPBrowser	www.wordpress.zzux.com
unknown 5	00591821f328911380277272164d08cd	6c260baa4367578778b1ecdaaab37ef9	Plugx	app.theworldfun.com cmc.apecscmc.com
unknown 6	0b3a047d31461e20887bb1d32b4e472f	21c46a95329f3f16050a7421841a92c4	downloader	mail.cbppnews.com
unknown 6	5392f1399a49935817669d22e5e644ea	b4522d05a9e3a034af481a7797a445ea	downloader	pic.nicklockluckydog.org
unknown 7	dceae0d1a680bc098bae9da466e12610	d6365edf2d3afa6d155273814b494eb3	PlugX	<varies>.qf.laoscript.org

Table 3: HT_Exploit payloads and actors

In Table 3, we can see that as many as nine distinct cyber espionage operators used CVE-2015-5119 exploits generated by the HT_Exploit kit.

The presence of the HT_Exploit kit generator is further confirmed by the two additional CVE-2015-5119 Flash files we discovered with the same creation date of 2015-07-07. As Table 4 illustrates these exploits had different ActionScript class names and instead of bundling the payloads internally both exploits download malicious payloads from remote servers. Additionally, as opposed to using LZMA compression these exploits used zlib compression. Further, the supporting ActionScript classes found in these files were different than the ActionScript classes seen above in the HT_Exploit files.

Actor	Date Created	SWF MD5	AS Class	CVE	Compression
Sofacy/APT28[8]	2015-07-07	557f8d4c6f8b386c32001def807dc715	Main	CVE-2015-5119	zlib
UPS/APT3[9]	2015-07-07	e9a57f70f739cb26dc053238b0a97425	MainClass	CVE-2015-5119	zlib

Table 4: CVE-2015-5119 outliers

The differences in the these two exploits versus the HT_Exploit samples indicates that neither Sofacy/APT28 nor the UPS/APT3 exploit were constructed with the same HT_Exploit generator tool.

This finding is significant as it offers evidence that the Sofacy/APT28 actor and UPS/APT3 actor maintain their own exploit supply chains or have in-house talent capable of exploit development. It is not a surprise that Sofacy, a cyber espionage operator believed to be based in Russia, does not share the same exploit supply chain as the actors using the HT_Exploit generator - many of whom are believed to be based in China.

---

8 http://www.welivesecurity.com/2015/07/10/sednit-apt-group-meets-hacking-team/

9 https://www.fireeye.com/blog/threat-research/2015/07/demonstrating_hustle.html

However, it is informative to gather definitive evidence that Chinese actors such as Wekby/APT18 and UPS/APT3 do not share the same exploit supply chain. This finding demonstrates that some Chinese-based operators employ differing intrusion techniques, tactics and procedures while also maintaining unique malware and exploit supply chains.

Cluster 2: flash_exploit_002

A second 0-day exploit targeting a previously unknown vulnerability in Adobe flash, CVE-2015-5122, was discovered from the Hacking Team archive and subsequently patched by Adobe on July 10, 2015.[10] Within one day of the release of a patch for this vulnerability multiple cyber espionage operators were observed sharing an exploit generator that bound this exploit to a payload of the operator's choice.

In total, we collected 11 Flash files that exploited CVE-2015-5122 and were created on July 11, 2015. All 11 of these files, as shown in Table 4, had the same ActionScript class name of flash_exploit_002, were compressed with the LZMA algorithm, and had payloads bundled with the malicious Flash file.

Date Created	SWF MD5	AS Class	CVE	Compression	Embedded Payload
7/11/2015	726bd0bd6cca8d481cf6165c95528caa	flash_exploit_002	CVE-2015-5122	LZMA	yes
7/11/2015	b65076f4cb6e74429dd02fcacda0bec3	flash_exploit_002	CVE-2015-5122	LZMA	yes
7/11/2015	8a8e9bbf1ca2a926f0a5d06217eeea55	flash_exploit_002	CVE-2015-5122	LZMA	yes
7/11/2015	054d9852de6983116bd3d521e8d73296	flash_exploit_002	CVE-2015-5122	LZMA	yes
7/11/2015	15112a53fcecc4c666a82ca84a853716	flash_exploit_002	CVE-2015-5122	LZMA	yes
7/11/2015	727dd4a7aae56a8202c5aa7758ea5d46	flash_exploit_002	CVE-2015-5122	LZMA	yes
7/11/2015	e33cf5b9f3991a8ee4e71f4380dd7eb1	flash_exploit_002	CVE-2015-5122	LZMA	yes
7/11/2015	451c52652ddb28e9071078f214a327a7	flash_exploit_002	CVE-2015-5122	LZMA	yes
7/11/2015	b1238ccbb10af3e81110d3afacd98161	flash_exploit_002	CVE-2015-5122	LZMA	yes
7/11/2015	b7d39c5833e5896b7f5849966095a4bf	flash_exploit_002	CVE-2015-5122	LZMA	yes
7/11/2015	d536c4b71d131848e965c4524780a8aa	flash_exploit_002	CVE-2015-5122	LZMA	yes

Table 4: flash_exploit_002 cluster

Furthermore, each of these 11 Flash files had identical ActionScript classes as shown in Table 5:

[10] https://helpx.adobe.com/security/products/flash-player/apsa15-04.html

ActionScript classname	MD5
bin_bin$943b2abb9578a8f4b0f6164ee413a25f648059697.as	1b127227d6228ce32b93d197756b6708
flash_exploit_002.as	b45bec70393db70c3c7c6d5f643cdd64
MyClass.as	785e8af0535717183f547b6d876513f0
MyClass32.as	00bdfdbc00dd1faa7896926b99444e2f
MyUtils.as	fa9142065d6550d729168b5977f2cf14
PayloadWin32.as	7d2e309c07099aaa2cf99d4075d77975
ShellWin32.as	026cb3d736b6cd7d3529e04e72d35923
test.as	0a28f677465fdf76689ca2fcabc68d53

Table 5: flash_exploit_002 ActionScript classes

In summary, The 11 Flash exploit files listed in Table 4 share the following characteristics:

- Created on the same date of 7/11/2015
- Targeted the same vulnerability of CVE-2015-5122
- Compressed with the LZMA algorithm
- Contained an embedded payload
- Had identical action script as shown in Table 5

Image 2: flash_exploit_002 cluster

These common attributes indicate that these malicious Flash files were created by a single exploit generator. As we observed with the HT_Exploit kit, the variety of payloads dropped by these flash_exploit_002 Flash files and command control infrastructure used by the dropped payloads suggest that different operators were using the same generator kit to create these Flash exploits.

Actor	SWF MD5	Payload MD5	Payload Family	C2 Server
wekby/APT18	726bd0bd6cca8d481cf6165c95528caa	80d234dc62c1bcec1466986f1224c205	gh0st (sycmentec)	223.25.233.248
unknown 7	054d9852de6983116bd3d521e8d73296	76808c0ade61f433bb5be83a4464eb9e	EvilGrab	inbox.webmailgoogle.com
unknown 1	b65076f4cb6e74429dd02fcacda0bec3	07aa0340ec0bfbb2e59f1cc50382c055	Emdivi	www.nichiiko-golf.com jp.virhub.biz www.n-fit-sub.com[11] www.sakuranorei.com
unknown 1	8a8e9bbf1ca2a926f0a5d06217eeea55	2a11d0f22b413d990437892ec6fb28a9	Emdivi	www.nichiiko-golf.com jp.virhub.biz www.n-fit-sub.com[12] www.sakuranorei.com
unknown 8	15112a53fcecc4c666a82ca84a853716	5e223ef669acd309697c90cac2f9953f	isspace	172.246.109.27
unknown 9	727dd4a7aae56a8202c5aa7758ea5d46	e43e14f6d1159ea9564bc23982b9afd5	PlugX	web.paramerat.com
unknown 10	e33cf5b9f3991a8ee4e71f4380dd7eb1	5a22e5aee4da2fe363b77f1351265a00	PlugX	amxil.opmuert.org
unknown 10	451c52652ddb28e9071078f214a327a7	5a22e5aee4da2fe363b77f1351265a00	PlugX	amxil.opmuert.org
unknown 11	b1238ccbb10af3e81110d3afacd98161	ebf157abfe656d87e43a63ca91507996	PlugX	211.226.71.4
unknown 12	b7d39c5833e5896b7f5849966095a4bf	6102f79567dff2168beb17aba31e058f	smac	whois.nictr.info
unknown 12	d536c4b71d131848e965c4524780a8aa	53fe5d10530fbef13da8c9e706a72944	smac	news.turkceil.tk

Table 6: flash_exploit_002 payloads and actors

Based on passive DNS analysis of the command and control infrastructure it appears that as many as eight different actors shared access to the flash_exploit_002 generator.

Cluster 3: exp1_fla

One day after the appearance of the exploits from the HT_Exploit generator were seen in the wild, a new set of exploits appeared targeting CVE-2015-5119. This new set of exploits contained ActionScript with the classname exp1_fla/MainTimeline. These malicious Flash files contained new functionality when compared to HT_Exploit.

[11] Note the www.n-fit-sub.com domain was used as command and control for a different Emdivi payload dropped by a CVE-2015-5119 exploit from the HT_Exploit cluster

[12] Ibid

Date Created	SWF MD5	AS Class	CVE	Compression	Embedded Payload
n/a	c101d289d36558c6fbe388d32bd32ab4	exp1_fla/MainTimeline	CVE-2015-5119	zlib	no
n/a	9bf3e6a95a261a449be02ac03d4f0523	exp1_fla/MainTimeline	CVE-2015-5119	zlib	no
n/a	4dd21fd277c772bcf8b9d1d72bf68de8	exp1_fla/MainTimeline	CVE-2015-5119	zlib	no
n/a	42b091f63548fccbbd87f8c06b632dda	exp1_fla/MainTimeline	CVE-2015-5119	none	yes
n/a	e15fb188c0c50d62657c7fd368a9a4ab	exp1_fla/MainTimeline	CVE-2015-5119	zlib	no
n/a	53473af71d40568d25da87fc41dfe500	exp1_fla/MainTimeline	CVE-2015-5119	zlib	no
n/a	5beb4504fe22e859a2b09cd5a654b23e	exp1_fla/MainTimeline	CVE-2015-5119	zlib	no

Table 7: exp1 cluster

A total of eight samples were collected, but the internal metadata from these files did not include timestamps. However, we collected last-modified dates for four of the SWF files. The dates ranged from 07/08/2015 through 07/14/2015.

These samples were different from exploits targeting CVE-2015-5119 seen in the HT_Exploit cluster in many ways. Samples seen in the exp1_fla cluster had the following properties:

- zlib compression
- remote payload retrieval
- verbose messages when viewed in a browser
- extra non-malicious AS

Alternate compression algorithms are not a big change over the previously observed kits. However, remote payload retrieval is a significant difference. This new feature allows the SWF's to be much smaller while also allowing the actors to switch out payloads on the server side over time.

Actor	SWF MD5	Remote Payload MD5	Payload Family	C2 Server
APT20	c101d289d36558c6fbe388d32bd32ab4	79f71f327a38c2226d36a21172d2922b	Poison Ivy	win7.myz.info
DNSCalc/APT12	9bf3e6a95a261a449be02ac03d4f0523	d6f7a1995a869dbd411c2b46364a6dc9	Ixeshe variant	95.110.210.31
DNSCalc/APT12	4dd21fd277c772bcf8b9d1d72bf68de8	87e01acad9b67953881c7d1b8e28d003	Ixeshe variant	70.90.107.24
unknown 13	42b091f63548fccbbd87f8c06b632dda		Linopid	twnic.ignorelist.com opp.jumpingcrab.com 220.134.9.49
unknown 14	e15fb188c0c50d62657c7fd368a9a4ab	1b47a8c22f9905afe05fad41ff3c9e4d	gh0st	yunwu1.xicp.net 2ph6wtzr4z.qstheory.org
unknown 15	53473af71d40568d25da87fc41dfe500	ec9f882d7eb9b60431e56ed4e25f3830	Plugx	news.voanews.hk
unknown 16	5beb4504fe22e859a2b09cd5a654b23e	b8ec26fcf2a4e855e04278f9bf5dc877	Unknown	eniw577dlcp4zbag.onion

Table 8: exp1 payloads and actors

As with both the HT_Exploit and flash_exploit_002 clusters, we observed multiple independent cyber espionage operators deploying exp1_fla exploits. However, unlike the HT_Exploit and flash_exploit_002 clusters we do not believe the exp1_fla exploits were built via a privately shared exploit generator. Instead, we believe the various operators seen in Table 8 shared exploit source code with one another.

The underlying ActionScript observed in the files from the exp1_fla cluster were not uniform across all 12 exploit files. Specifically, as shown in Table 9, the MyClass ActionScript varied from files to file within the exp1_fla cluster. Also, note that the MyClass1 ActionScript class appears to have been borrowed from Metasploit.

AS Classname	MD5 Comparison
exp1_fla/MainTimeline.as	same for all 12 files
MyClass.as	different
MyClass1.as	same for all 12 files[13]
MyClass2.as	same for all 12 files
MyUtils.as	same for all 12 files

Table 9: exp1_fla ActionScript comparison

It is therefore unlikely that the files seen in the exp1_fla cluster were created via a shared exploit generator. A single generator would be unlikely to produce the differences seen in the underlying ActionScript. However, it is also doubtful that the above four underlying ActionScript classes seen in Table 9 would be identical unless the different operators were sharing code.

Cluster 4: exp2_fla

Only two malicious files targeting CVE-2015-5122 with the main ActionScript class name of exp2_fla were observed in the wild. Unlike the exploits observed in the flash_exploit_002 cluster, the exploits in the exp2_fla cluster were zlib compressed and payloads were downloaded from remote servers.

As evidenced by their respective Last-Modified dates, both of the files from the exp2_fla cluster were first deployed in the wild on July 14, 2015. The first file from the flash_exploit_002 cluster seen in the wild was observed on July 12, 2015. This indicates that the flash_exploit_002 generator was available to cyber espionage operators prior to the availability of the exploit code seen in the exp2_fla cluster.

Date Created	SWF MD5	AS Class	CVE	Compression	Embedded Payload
n/a	195bdc84f114c282e61f206dc88cd26d	exp2_fla/MainTimeline	CVE-2015-5122	zlib	no
n/a	aaa62d5f0e348f0e890ad9d3f71e448d	exp2_fla/MainTimeline	CVE-2015-5122	zlib	no

Table 10: exp2 cluster

[13] This ActionScript class was seen in the metasploit module targeting CVE-2015-5119

An analysis of the malicious ActionScript found in both 195bdc84f114c282e61f206dc88cd26d and aaa62d5f0e348f0e890ad9d3f71e448d reveal that the exploit code was not the same.

Sample One: 195bdc84f114c282e61f206dc88cd26d AS Class MD5	AS Class	Sample Two: aaa62d5f0e348f0e890ad9d3f71e448d AS Class MD5
3e7f8f4f2fdd7c587d0212ad38c10805	MyClass.as	058fe24b7de10d915737ede604b3954e
3614e902f822b6c30e024b80e7f1487b	MyClass32.as	3614e902f822b6c30e024b80e7f1487b
4eaa236e48598bce7e9b67edb143ca79	MyClass64.as	4eaa236e48598bce7e9b67edb143ca79
3fa797e193ff815afc9378c3a025bcde	MyUtils.as	76bbf9cfe6d6870d3e35cf038c39234c
504eedb7ed01bc7748d2bdaf7f0e48cc	exp2_fla/MainTimeline.as	504eedb7ed01bc7748d2bdaf7f0e48cc
acf3b75887d85dcc046792fd83664ef6	ShellMac32.as	acf3b75887d85dcc046792fd83664ef6
b067468484fa4fc1bb27a1a4dcead881	ShellMac64.as	b067468484fa4fc1bb27a1a4dcead881
2ad0335cc530ebfe59901e4d3b31db7b	ShellWin32.as	2ad0335cc530ebfe59901e4d3b31db7b
e1cd6400f115f60213764347f927f7e6	ShellWin64.as	e1cd6400f115f60213764347f927f7e6

Table 11: exp2 ActionScript classes

Table 11 illustrates that while some of the ActionScript classes were shared by the two different operators observed using exploit code from the exp2_fla cluster, other classes had been modified by the different operators. The data in Table 12 shows the payloads used by the two different actors.

Actor	SWF MD5	Payload MD5	Payload Family	C2 Server
APT20	195bdc84f114c282e61f206dc88cd26d	bdc263c93bc5bd0d31a517be469a697a	Poison Ivy	win7.myz.info
unknown 17	aaa62d5f0e348f0e890ad9d3f71e448d	d22f5f14f573293231f04cc53fee17f9	Poison Ivy	jiussharefiles.ddns.net fileshare.serveftp.com

Table 12: exp2 payloads and actors

This data suggests that the APT20 and Unknown 17 actor were not sharing a generator tool. Rather, it appears that these actors were sharing exploit source code and modifying this code to suit their own individual needs. It is unlikely that a single generator would produce the differences seen in the underlying ActionScript. However, it is also unlikely that five of the underlying ActionScript classes would be identical unless the different operators were sharing code or tools.

Cluster 5: movie_fla

Unlike the other clusters of activity documented above, the files in the movie_fla cluster were deployed by a single actor. This actor is known as DNSCalc/APT12. The payloads downloaded by the exploits seen in the table below appear to be variants of the Ixeshe malware family - a tool previously used by this actor.[14]

Date Created	SWF MD5	AS Class	CVE	Compression	Embedded Payload
n/a	edcd313791506c623d8a2a88b9b0e84c	movie_fla/MainTimeline	CVE-2015-5119	zlib	no
n/a	83388058055d325a2fa5288182a41e89	movie_fla/MainTimeline	CVE-2015-5119	zlib	no
n/a	aa9eded1eb95f026aaf84919cc27ad32	movie_fla/MainTimeline	CVE-2015-5119	zlib	no

Table 13: movie ActionScript classes

Both edcd313791506c623d8a2a88b9b0e84c and 83388058055d325a2fa5288182a41e89 pull down the same payload from 213.186.164.211/news/in.gif. These payloads are encoded with a single-byte XOR key. The decoded payload had a MD5 of 4dfdfd203eeeff75474b8f431b6e0750.

The third sample, aa9eded1eb95f026aaf84919cc27ad32 downloaded a payload from 84.124.26.234/image/welcome.gif. The payload is also encoded with a single-byte XOR key. This decoded payload had an MD5 of 5dd963d33c31cdb9131d86241e754d81.

The movie_fla cluster is of note not only because a single actor deployed files from this clusters, but also because the exploit code from the movie_fla cluster was derived from the exp1_fla cluster. Table 14 demonstrates the DNSCalc/APT12 operator used code from the exp1_fla cluster to develop the exploits in the movie_fla cluster.

SWF MD5	Main AS Class name	MyClass1.as MD5	MyClass2.as MD5
aa9eded1eb95f026aaf84919cc27ad32	movie_fla/MainTimeline	4b705980ed1b07becd76f47e007b5b3a[15]	34b614df1e57f2ce95997f85078de2f9
9bf3e6a95a261a449be02ac03d4f0523	exp1_fla/MainTimeline	4b705980ed1b07becd76f47e007b5b3a[16]	34b614df1e57f2ce95997f85078de2f9
4dd21fd277c772bcf8b9d1d72bf68de8	exp1_fla/MainTime	4b705980ed1b07becd76f47e007b5b3a[17]	34b614df1e57f2ce95997f85078de2f9

Table 14: DNSCalc/APT12 Exploit Development across clusters

Finally, the variation of the ActionScript classes within the movie_fla cluster suggest that the DNSCalc/APT12 operator continued to modify the exploit code for the purposes of changing the remote url for payload download.

SWF MD5	Main AS Class name	Shellwin32.as MD5
edcd313791506c623d8a2a88b9b0e84c	movie_fla/MainTimeline	541f6853cef8144574d8fcdb89aef9e1
83388058055d325a2fa5288182a41e89	movie_fla/MainTimeline	541f6853cef8144574d8fcdb89aef9e1
aa9eded1eb95f026aaf84919cc27ad32	movie_fla/MainTimeline	8e52606b6c31f27b5984ac086f8c0b0f

Table 15: DNSCalc/APT12 Exploit Development within the movie_fla

[14] https://www.fireeye.com/blog/threat-research/2014/09/darwins-favorite-apt-group-2.html
[15] This ActionScript class was seen in the metasploit module targeting CVE-2015-5119
[16] Ibid
[17] Ibid

These findings suggests that DNSCalc/APT12 maintains access to either an exploit research and development supply chain.

Trend Analysis

Viewed in total, this data presents an interesting picture and highlights potential relationships between independent cyber espionage operators. The timeline presented in Image 1 illustrates when operators deployed exploits from each of the clusters discussed above.

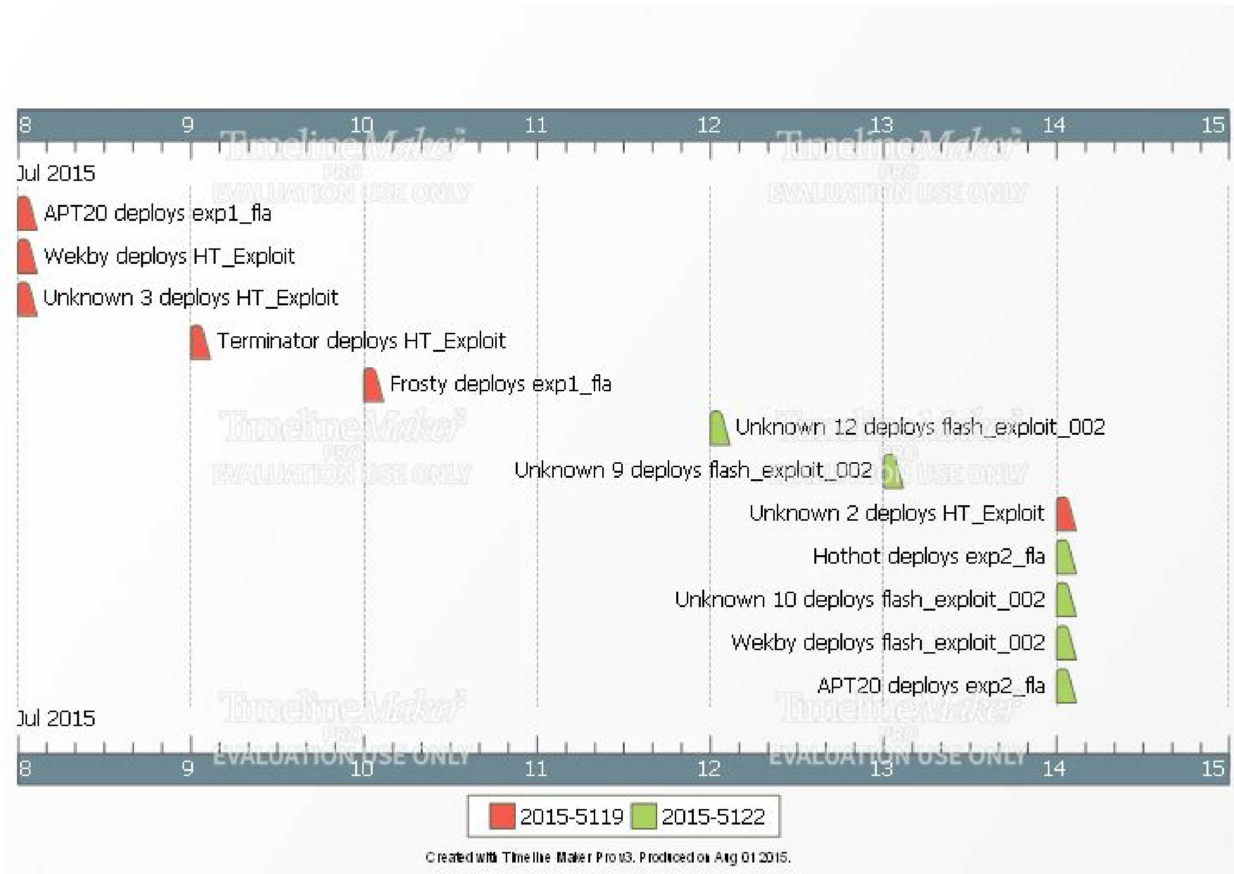

Image 3: Exploit Deployment Timeline

The above timeline represents only a portion of our entire data set because we were not in all cases able to collect the Last-Modified date from a source in the wild. However, with the subset of data that we collected we can reach two conclusions.

First, the operators known as Wekby/APT18 and APT20 quickly deployed both CVE-2015-5119 and CVE-2015-5122 exploits. Both Wekby/APT18 and APT20 deployed CVE-2015-5119 and CVE-5122 exploits on 7/8/2015 and 7/14/2015 respectively. In the case of 2015-5119, Wekby/APT18 and APT20 deployed their malicious files on the first day that the exploits were seen in the wild. In the case of CVE-2015-5122, both Wekby/APT18 and APT20 deployed their exploits only two days after other groups were seen using CVE-2015-5122 in the wild.

Note that Wekby/APT18 and APT20 were not sharing either the HT_Exploit nor flash_exploit_002 generators. The differences in the exploits used by both of the groups illustrates that the operators were not sharing code or tools. The differences are seen in the ActionScript, the compression, and location of the dropped payloads.

Therefore it does not appear that the two operators collaborated, but rather both maintain their own independent logistic operations. The speed at which both of the groups acquired either an exploit generator or source code suggests that they are both well connected to existing supply chains or maintain their own in-house talent capable of discovering vulnerabilities and developing exploit code.

Second, multiple operators appear to have access to their own unique exploit research and development supply chains or in-house talent. Groups that exhibited this characteristic include Wekby/APT18, APT20, UPS/APT3, and DNSCalc/APT12. Each of the groups were observed using unique ActionScript. In the case of APT20, other operators were observed using variations of the same exp1_fla ActionScript, but in that case APT20 was the first operator observed using ActionScript from the exp1_fla cluster in the wild.

Analysis of Competing Hypotheses

The data presented indicates that independent cyber espionage operators share exploit generators or code amongst themselves. This finding supports previous research that demonstrates that actors also share attack tools.[18]

While it is evident that independent operators are sharing exploit generators and code, the structure of these sharing relationships is unclear. There are several explanations to describe how two different exploit generators and similar code were distributed to several different attackers.

A Single Quartermaster

It is possible that there is a single entity responsible for vulnerability research and exploit development. This organization's mission would be to discover 0-day exploits, produce weaponized code, and develop generators. These generators would then be shared with independent cyber espionage operators for use in different campaigns.

This structure would explain the usage patterns observed with HT_Exploit and flash_exploit_002 generators. Multiple independent operators were observed using exploits derived from these generators in distinct campaigns.

---

[18] https://www.fireeye.com/content/dam/fireeye-www/global/en/current-threats/pdfs/rpt-malware-supply-chain.pdf

It is unlikely that several different operators would be able to produce multiple exploit files with identical ActionScript without the use of a shared generator.

Formal or Informal Sharing of Tools

Another plausible explanation for the observed sharing of exploit generators and code discussed within this paper is that one or a small number of cyber espionage operators maintain their own exploit supply chain or in-house research and development capability. As these operators develop tools, they either formally or informally share the fruits of their labor with other affiliated groups.

For example, it appears that Wekby/APT18, a more established operator, either has access to its own exploit supply chain or has direct access to individuals with a background in vulnerability research and exploit development. In this proposed example Wekby/APT18 could then share the generator developed for their own operators with operators from other distinct groups.

While this structure of individual exploit supply chains or access to in house talent may not fully explain the patterns of sharing seen across the five clusters of CVE-2015-5119 and CVE-2015-5122 exploits, it is evident that this model of exploit development does exist in some cases. In addition to Wekby/APT18, this model was also seen with UPS/APT3's and Sofacy/APT28's use of ActionScript code different than the exploits created by the HT_Exploit generator on 2015-07-07. The use of different ActionScript indicates that UPS/APT3 and Sofacy/APT28 implemented their own versions of CVE-2015-5119, while Wekby/APT18 and others used a shared generator to produce their exploits.

Formal or Informal Sharing of Code

As opposed to the model of formal or informal sharing of tools, a model of formal or informal sharing of code suggests that operators individually maintain their own in-house exploit research and development capability and only share code fragments or basic knowledge amongst themselves.

In this model fully functioning generators are not shared, but instead classes or other code snippets may be shared. This model could explain the sharing patterns observed in the exp1_fla, exp2_fla, and movie_fla clusters of CVE-2015-5119 and CVE-2015-5122 exploits.

Conclusion

It is unlikely that any one hypothesis by itself can fully explain the data presented in this paper. The first model of a single quartermaster can be used to explain the patterns observed in the HT_Exploit and flash_exploit_002 clusters, but fail to explain the patterns in the exp1_fla, exp2_fla, and movie_fla clusters.

The model of a single quartermaster developing and sharing generators would explain the identical nature of the malicious ActionScript classes in the HT_Exploit and flash_exploit_002 clusters. However, the small variations in the ActionScript classes seen in the exp1_fla, exp2_fla, and movie_fla clusters suggest that a generator was not used to produce the Flash exploits seen in those clusters. It is unlikely that a single generator would produce the minor differences observed in a subset of the ActionScript classes seen in the exp1_fla, exp2_fla and movie_fla clusters.

The second model, where a subset of operators either acquire or build then subsequently share tools with other less capable actors may also explain the HT_Exploit and flash_exploit_002 clusters but does not appear to sufficiently account for the exp1_fla, exp2_fla, and movie_fla clusters.

The model of a subset of operators that are capable of producing and then sharing generators would explain the matching ActionScript seen in the HT_Exploit and flash_exploit_002 clusters. However, this model cannot explain the minor differences seen in the exp1_fla, exp2_fla and movie_fla clusters as it is unlikely that a common generator would produce the observed disparity in the ActionScript.

Finally, the model where each operator maintains their own exploit supply chain or in-house exploit research and development capability and in turn share classes or code snippets amongst themselves may explain the exp1_fla, exp2_fla and movie_fla clusters but does not adequately account for the HT_Exploit or flash_exploit_002 clusters.

This model would explain the minor variations seen in the ActionScript found in the exp1_fla, exp2_fla and movie_fla clusters, but it is unlikely that it could explain the uniformity in the ActionScript seen in the HT_Exploit and flash_exploit_002 clusters.

	One Quartermaster	Shared Generators	Shared Code
HT_Exploit	valid explanation	valid explanation	invalid explanation
flash_exploit_002	valid explanation	valid explanation	invalid explanation
exp1_fla	invalid explanation	invalid explanation	valid explanation
exp2_fla	invalid explanation	invalid explanation	valid explanation
movie_fla	invalid explanation	invalid explanation	valid explanation

Table 15: Competing Hypotheses

As a result, it is likely that a mix of these models can be used to explain the data presented in this paper. Therefore, through the data presented we can conclude that independent cyber espionage operators maintain a complex set of either formal or informal relationships that govern how these actors develop and share tools, code and tactics.