FireEye
SECURITY REIMAGINED

SPECIAL REPORT

FIREEYE THREAT INTELLIGENCE

HAMMERTOSS:
Stealthy Tactics Define a Russian Cyber Threat Group

JULY 2015

SECURITY REIMAGINED

CONTENTS

HAMMERTOSS 3

APT29 5

Introducing HAMMERTOSS 6
Five Stages of HAMMERTOSS 6

Stage 1: The Communication Process Begins with Twitter 7
Figure 1: HAMMERTOSS calls out to a Twitter handle 7

Stage 2: Tweeting a URL, Minimum File Size of an Image, and Part of an Encryption Key 8
Figure 2: Learning the URL, image size, and encryption key 8
Figure 3: Twitter page for d3109c83e07dd5d7fe032dc80c581d08 9

Stage 3: Visiting GitHub to Download an Image 10
Figure 4: The active Twitter account in our sample contained a GitHub URL and a related GitHub page with image containing encrypted data 10

Stage 4: APT29 Employs Basic Steganography 11
Figure 5: Encrypted data appended beyond the `FF D9` JPEG End of File marker 11

Stage 5: Executing Commands and Uploading Victim Data 12
Figure 6: Executing Commands and Removing Data 12

Conclusion 13
Difficulty Identifying Accounts, Discerning Legitimate and Malicious Traffic, and Predicting the Payload

APT29: An Adaptive and Disciplined Threat Group 13

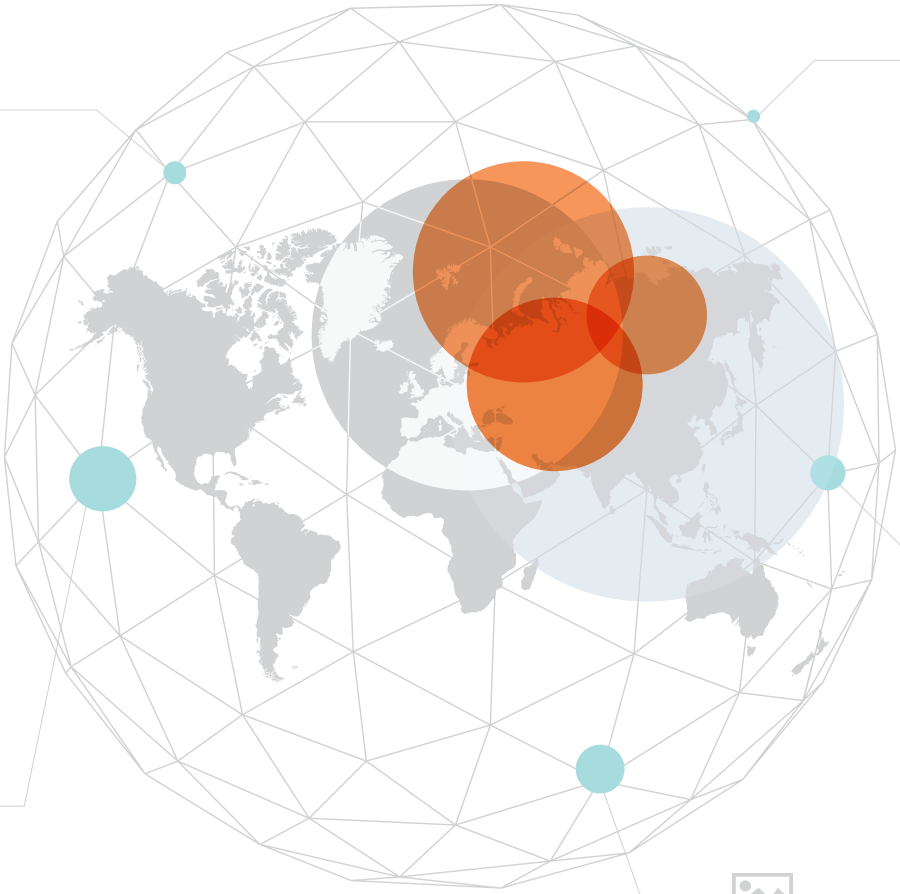
HAMMERTOSS

The Russian cyber threat groups that we monitor frequently design innovative ways to cover their tracks. In early 2015, we came across stealthy malware—which we call HAMMERTOSS—from an advanced persistent threat group that we suspect the Russian government sponsors. We designate this group APT29.

Using a variety of techniques—from creating an algorithm that generates daily Twitter handles to embedding pictures with commands—the developers behind HAMMERTOSS have devised a particularly effective tool. APT29 tries to undermine the detection of the malware by adding layers of obfuscation and mimicking the behavior of legitimate users. HAMMERTOSS uses Twitter, GitHub, and cloud storage services to relay commands and extract data from compromised networks.

Using a variety of techniques—from creating algorithms that generate daily Twitter handles to embedding pictures with commands—the developers behind HAMMERTOSS have devised a particularly effective tool.

HAMMERTOSS works by:



1. Retrieving commands via legitimate web services, such as Twitter and GitHub, or using compromised web servers for command-and-control (CnC).

2. Visiting different Twitter handles daily and automatically.

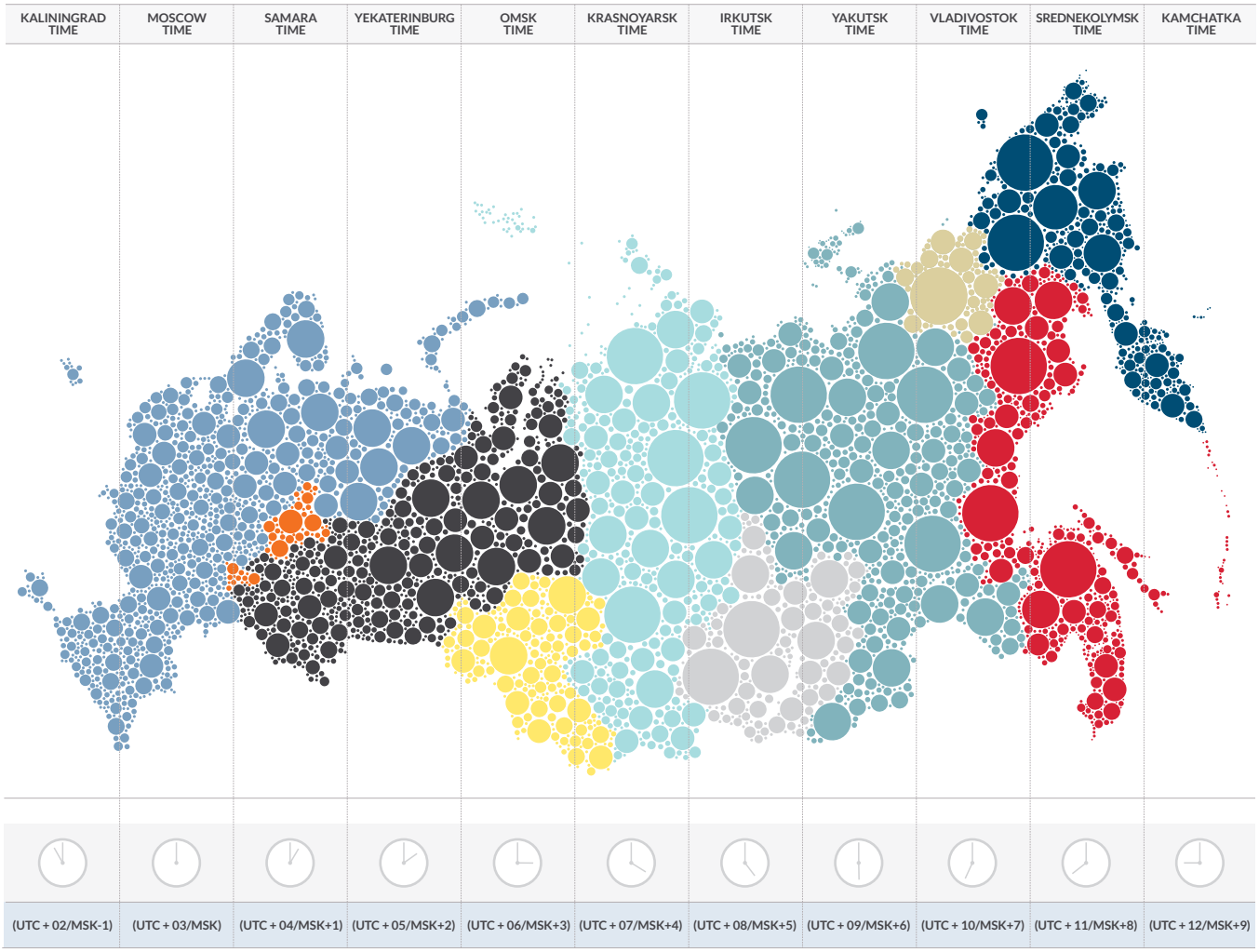
3. Using timed starts—communicating after a specific date or only during the victim's workweek.

4. Obtaining commands via images containing hidden and encrypted data.

5. Extracting information from a compromised network and uploading files to cloud storage services.

While none of these tactics are new, the combination of these techniques piqued our interest.

APT29



APT29 has been operating in its current form since at least late 2014. We suspect the Russian government sponsors the group because of the organizations it targets and the data it steals. Additionally, APT29 appeared to cease operations on Russian holidays, and their work hours seem to align with the UTC +3 time zone, which contains cities such as Moscow and St. Petersburg.

While other APT groups try to cover their tracks to thwart investigators, very few groups show the same discipline and consistency.

Similarly, few groups display the ability to adapt to network defenders' attempts to mitigate its activity or remove it from victim networks. For example, APT29 almost always uses anti-forensic techniques, and they monitor victim remediation efforts to subvert them. Likewise, the group appears to almost solely uses compromised servers for CnC to enhance the security of its operations and maintains a rapid development cycle for its malware by quickly modifying tools to undermine detection. These aspects make APT29 one of the most capable APT groups that we track.

INTRODUCING HAMMERTOSS

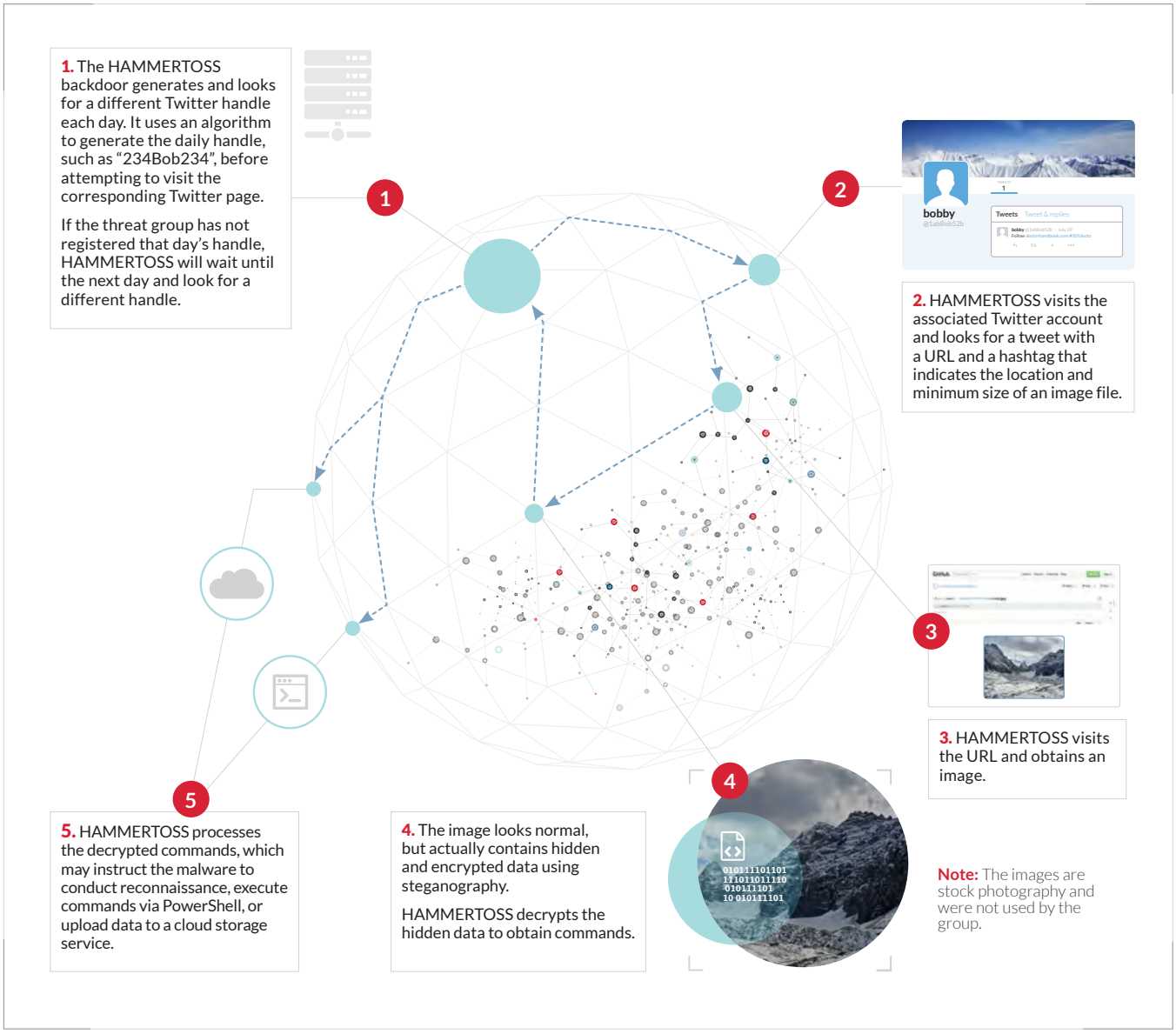
We first identified HAMMERTOSS in early 2015. APT29 likely used HAMMERTOSS as a backup for their two primary backdoors to execute commands and maintain access if the group's principal tools were discovered. We have identified two HAMMERTOSS variants that give APT29 alternative ways to communicate with the malware. The developer appears to name these variants *uploader* and *tDiscoverer*.[1] Both variants are written in the C# programming language. Each variant uses different methods to acquire CnC instructions, either by directly accessing a hard-coded website or accessing Twitter as an intermediary.

- *Uploader* is preconfigured to use a hard-coded server for its CnC. It goes to a specific URL to obtain an image with a specific file size.
- *tDiscoverer* uses an additional layer of obfuscation by first going to Twitter to obtain a CnC URL, before visiting the URL to acquire its target image.

We will focus on *tDiscoverer* in this report.

Five Stages of HAMMERTOSS

We have broken down the malware communication process into five stages to explain how the tool operates, receives instructions, and extracts information from victim networks. The stages include information on what APT29 does outside of the compromised network to communicate with HAMMERTOSS and a brief assessment of the tool's ability to mask its activity.



[1] The "*tDiscoverer*" variants were originally named "tDiscoverer.exe," and the "*Uploader*" variants had a debug path containing "uploader.pdb."

STAGE 1:
The Communication Process Begins with Twitter

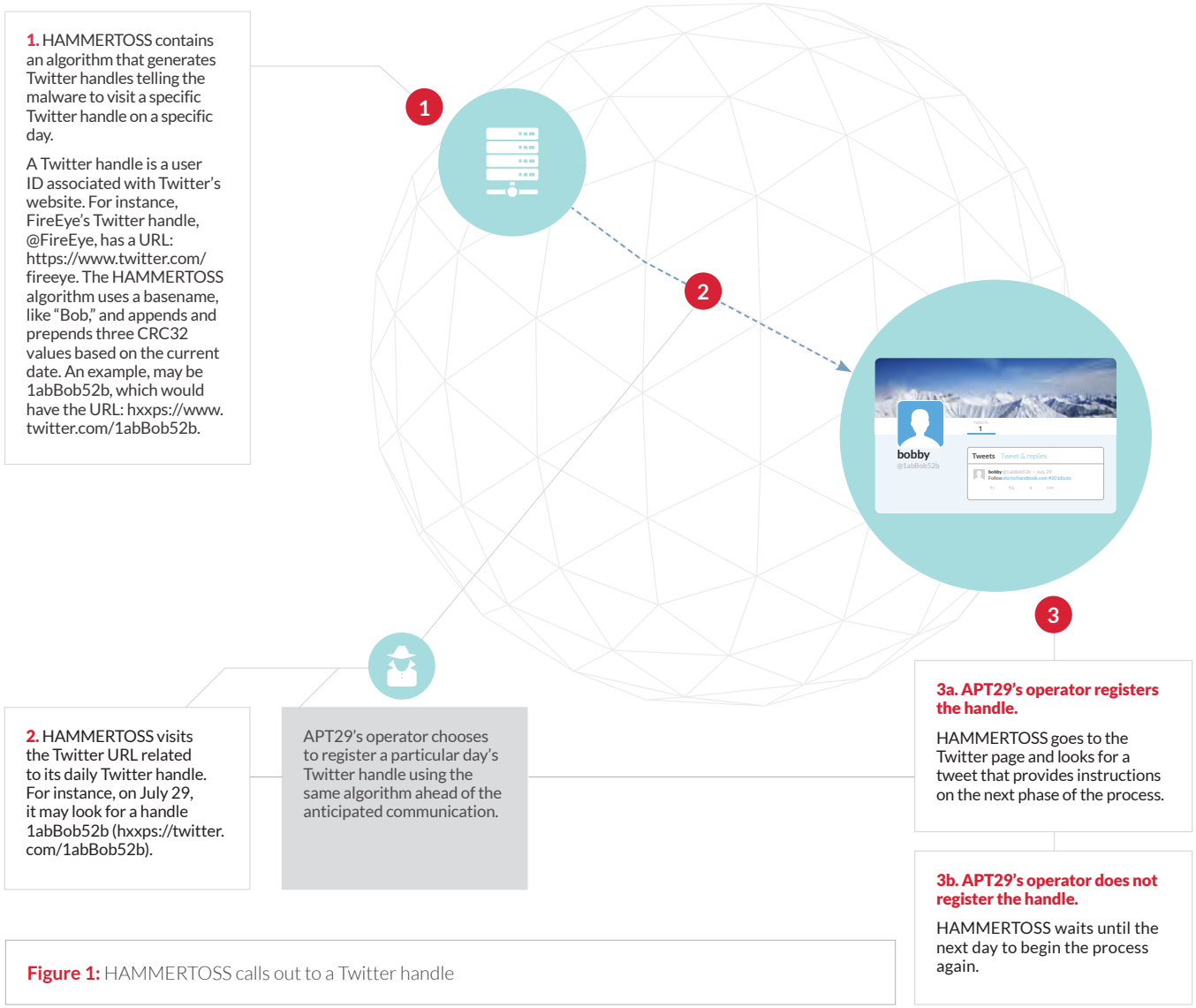


Figure 1: HAMMERTOSS calls out to a Twitter handle

HAMMERTOSS first looks for instructions on Twitter.

The malware contains an algorithm that generates a daily Twitter handle, which is an account user ID. To create the handles, the algorithm employs a basename, such as "Bob," and appends and prepends three CRC32 values based on the date. For example, "1abBob52b" would have the URL: hxxps://twitter.com/1abBob52b. Each HAMMERTOSS sample will create a different Twitter handle each day.

APT29 knows the algorithm used to generate the handles and chooses to register a Twitter handle and post obfuscated instructions to the handle's URL before the malware attempts to query it. If a particular day's handle is not registered and the URL for that day is not found, HAMMERTOSS will wait until the next day to attempt to communicate with another handle.

APT29 typically configures HAMMERTOSS to communicate within certain restrictions, such as only checking the Twitter handle on weekdays or after a specified start date. This allows the malware to blend in to "normal" traffic during the victim's work week or to remain dormant for a period of time before activating.

STAGE 2:
Tweeting a URL, Minimum File Size of an Image, and Part of an Encryption Key

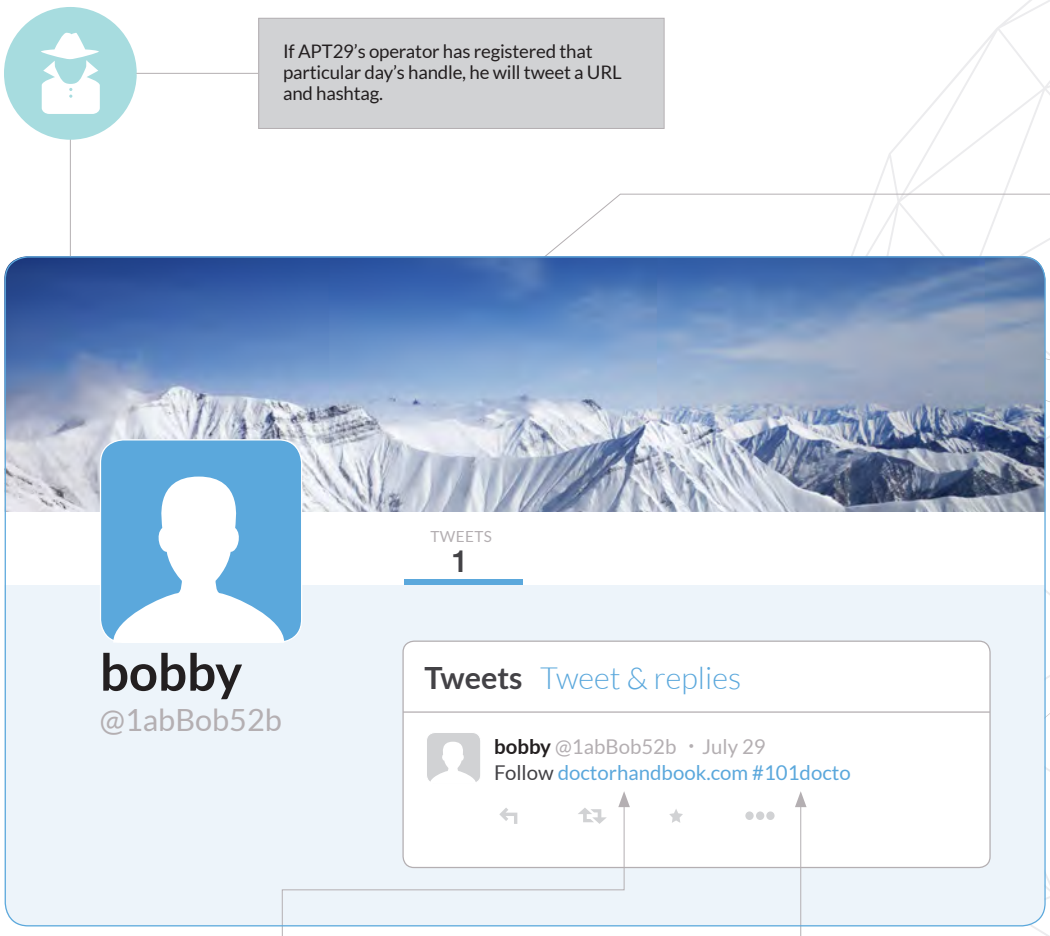


Figure 2: Learning the URL, image size, and encryption key

If APT29 has registered that day's Twitter handle, they will tweet a URL and a hashtag. The URL directs HAMMERTOSS to a webpage containing an image or images. The hashtag provides a number representing a location within the image file and characters for appending to an encryption key to decrypt instructions within the image. In the mockup of a HAMMERTOSS tweet in Figure 2, the hashtag was #101docto, indicating that the encrypted data begins at an offset of 101 bytes into the image file, and the characters *docto* should be added to the encryption key to decrypt the data.

Using Twitter as an intermediary to deliver the second-stage CnC to HAMMERTOSS allows APT29 to dynamically direct the tool.

In Figure 3 is a sample of the HAMMERTOSS *tDiscoverer* variant and a corresponding snapshot of a Twitter account page from one of its generated handles. At the time of publication, a publicly available HAMMERTOSS sample had only five generic detections in VirusTotal. The Twitter account was active and contained a link to a website.

MD5:	d3109c83e07dd5d7fe032dc80c581d08 (VirusTotal)
SHA1:	42e6da9a08802b5ce5d1f754d4567665637b47bc
Timing Behavior:	Communicate on weekdays only after April 3, 2015
Active Twitter Handle:	twitter[.]com/3c6Diallo7f0 (Figure 3 below)
Tweeted URL, Hashtag:	hxxp://www[.]doctorhandbook[.]com, #101docto
Detection Ratio:	5/56
Metadata:	**LegalCopyright**: © Microsoft Corporation. All rights reserved. **InternalName**: WerMgr **FileVersion**: 6.1.7600.16385 (win7_rtm.090713-1255) **CompanyName**: Microsoft Corporation **ProductName**: Microsoft® Windows® Operating System **ProductVersion**: 6.1.7600.16385 **FileDescription**: Windows Problem Reporting **OriginalFilename**: WerMgr

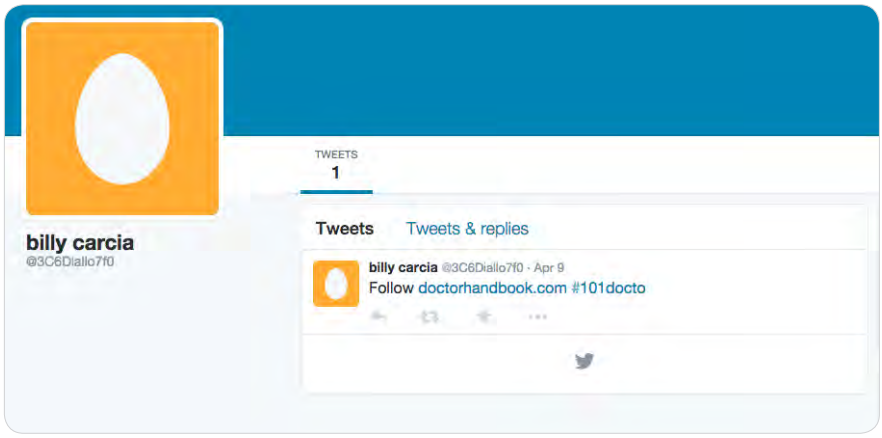


Figure 3: Twitter page for d3109c83e07dd5d7fe032dc80c581d08

HIDING AMONG UNREGISTERED TWITTER ACCOUNTS

HAMMERTOSS uses an algorithm to generate hundreds of Twitter handles annually for potential CnC. Many of these are unregistered, as APT29 chooses to register a particular day's handle as needed and ahead of an anticipated HAMMERTOSS beacon. This small number of registered accounts allows the group to maintain a small footprint.

Other tools use Twitter to relay instructions, including:[2]

- MiniDuke, a Windows-based backdoor that is a suspected Russian tool
- the Sninfs botnet
- Flashback, a Mac-based backdoor

MiniDuke behaves similarly to HAMMERTOSS by not only using Twitter for CnC, but also by downloading image files containing encrypted, appended content.

[2]"Miniduke still duking it out." ESET Security. 20 May 2014. http://www.welivesecurity.com/2014/05/20/miniduke-still-duking/ Balazs, Biro, Christian Istrate, and Mairus Tivaradar. A Closer Look at MiniDuke. BitDefender. 2013. http://labs.bitdefender.com/wp-content/uploads/downloads/2013/04/MiniDuke_Paper_Final.pdf. James, Peter. Flashback Mac Malware Uses Twitter as Command and Control Center. Intego's The Mac Security Blog. 5 March 2012. http://www.intego.com/mac-security-blog/flashback-mac-malware-uses-twitter-as-command-and-control-center. Coogan, Peter. "Twittering Botnets." Symantec Security Blog. 14 Aug 2009. http://www.symantec.com/connect/blogs/twittering-botnets. Kessler, Michelle. "Hackers harness Twitter to do their dirty work." USA Today. 17 August 2008. http://content.usatoday.com/communities/technologylive/post/2009/08/68497133/1#.VbJVi4q9_Vs.

STAGE 3:
Visiting GitHub to Download an Image

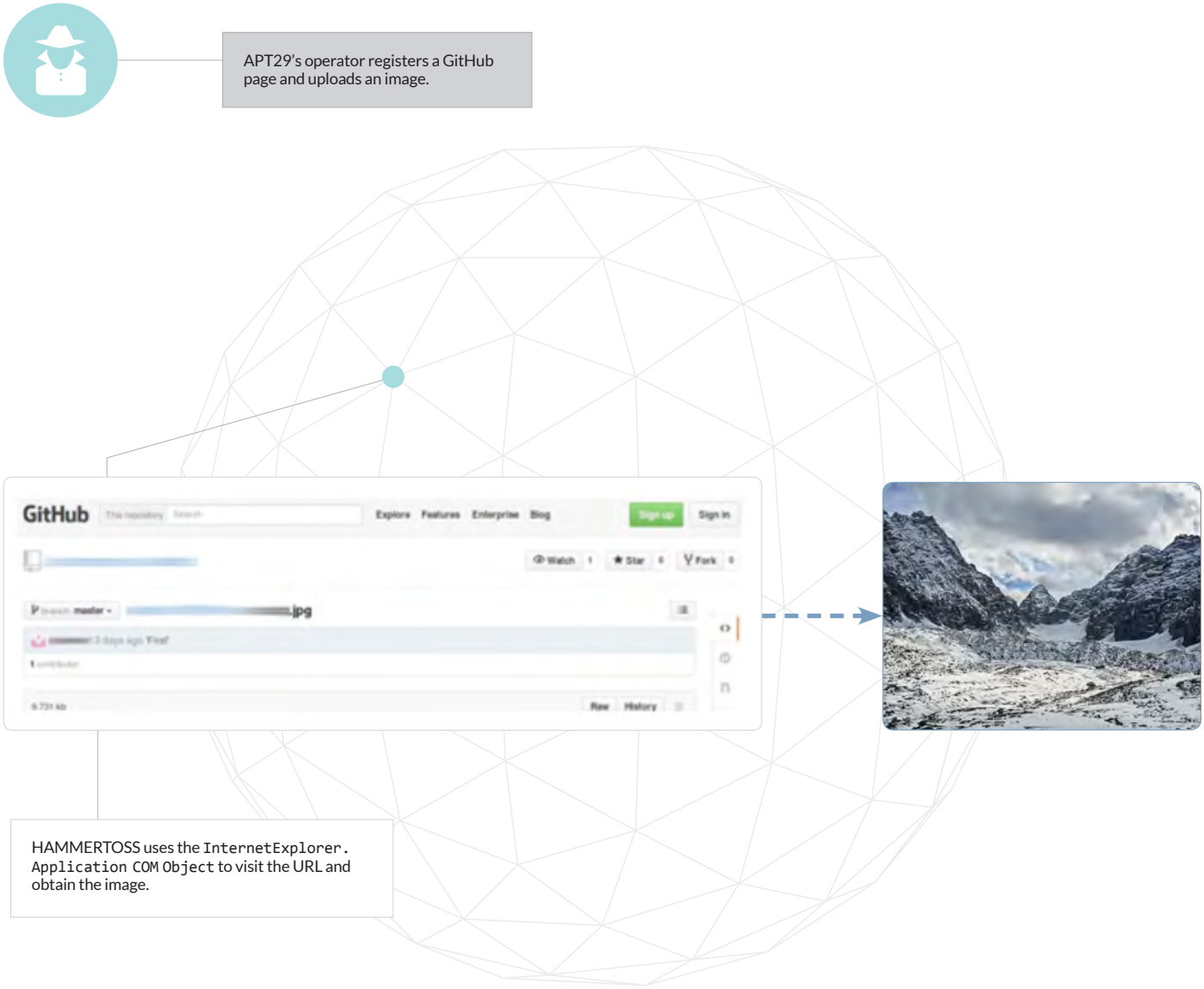
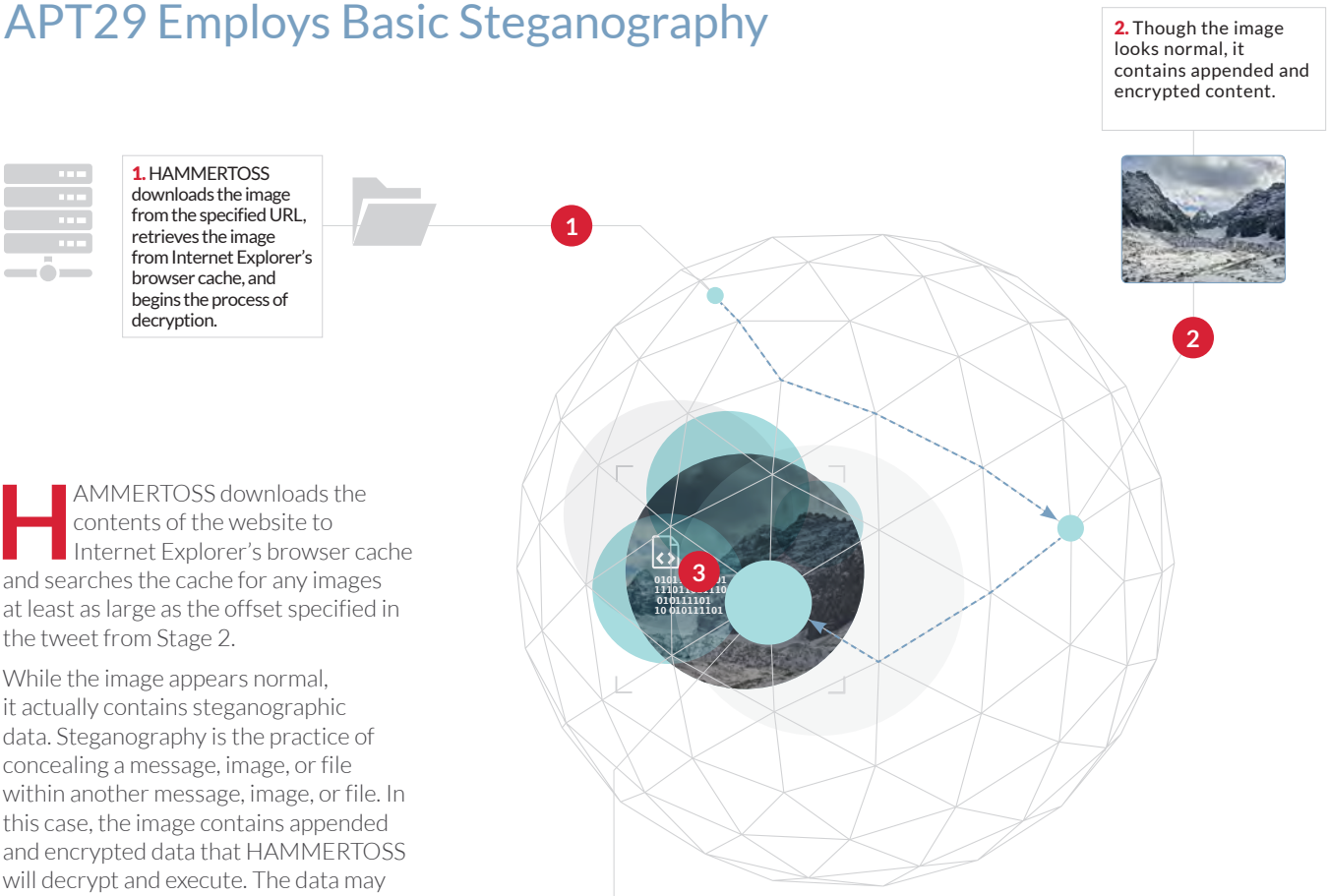


Figure 4: The URL specified in the tweet (in this case, a GitHub page) contains an image with appended and encrypted data

HAMMERTOSS then uses the `InternetExplorer.Application COM Object` to visit the URL specified in a tweet. We have observed URLs lead to specific GitHub accounts or compromised websites.

We will use Github for the next part in our example. Once HAMMERTOSS obtains the GitHub URL from its daily Twitter account, it visits the URL and downloads the contents of the page, including any image files.

STAGE 4:
APT29 Employs Basic Steganography



HAMMERTOSS downloads the contents of the website to Internet Explorer's browser cache and searches the cache for any images at least as large as the offset specified in the tweet from Stage 2.

While the image appears normal, it actually contains steganographic data. Steganography is the practice of concealing a message, image, or file within another message, image, or file. In this case, the image contains appended and encrypted data that HAMMERTOSS will decrypt and execute. The data may include commands or login credentials to upload a victim's data to a cloud storage service. HAMMERTOSS locates the encrypted data at the offset specified in the tweet in Stage 2. It decrypts the data using a key comprised of hard-coded data from the malware binary appended with the characters from the tweet.

Figure 5: Encrypted data appended beyond the `FF D9` JPEG End of File marker

APT29 ADDING STEGANOGRAPHY AS ANOTHER LAYER OF OBFUSCATION

We have observed only a few APT groups using steganography. HAMMERTOSS uses steganography by appending data to an image file after the image's end of file marker. This technique would be readily detectable if someone was checking for it. However, the appended data is encrypted, so even if detected, the investigator would be unable to decrypt the data without key material from two sources: the malware binary and the current tweet.

Indicative of APT29's discipline, the group ensures that if network defenders discover the images, the defenders still require the malware sample, corresponding Twitter handle, and tweet with the additional key material to decrypt the tool's instructions. All of the samples we have observed have used different encryption keys to decrypt the appended content.

STAGE 5:
Executing Commands and Uploading Victim Data

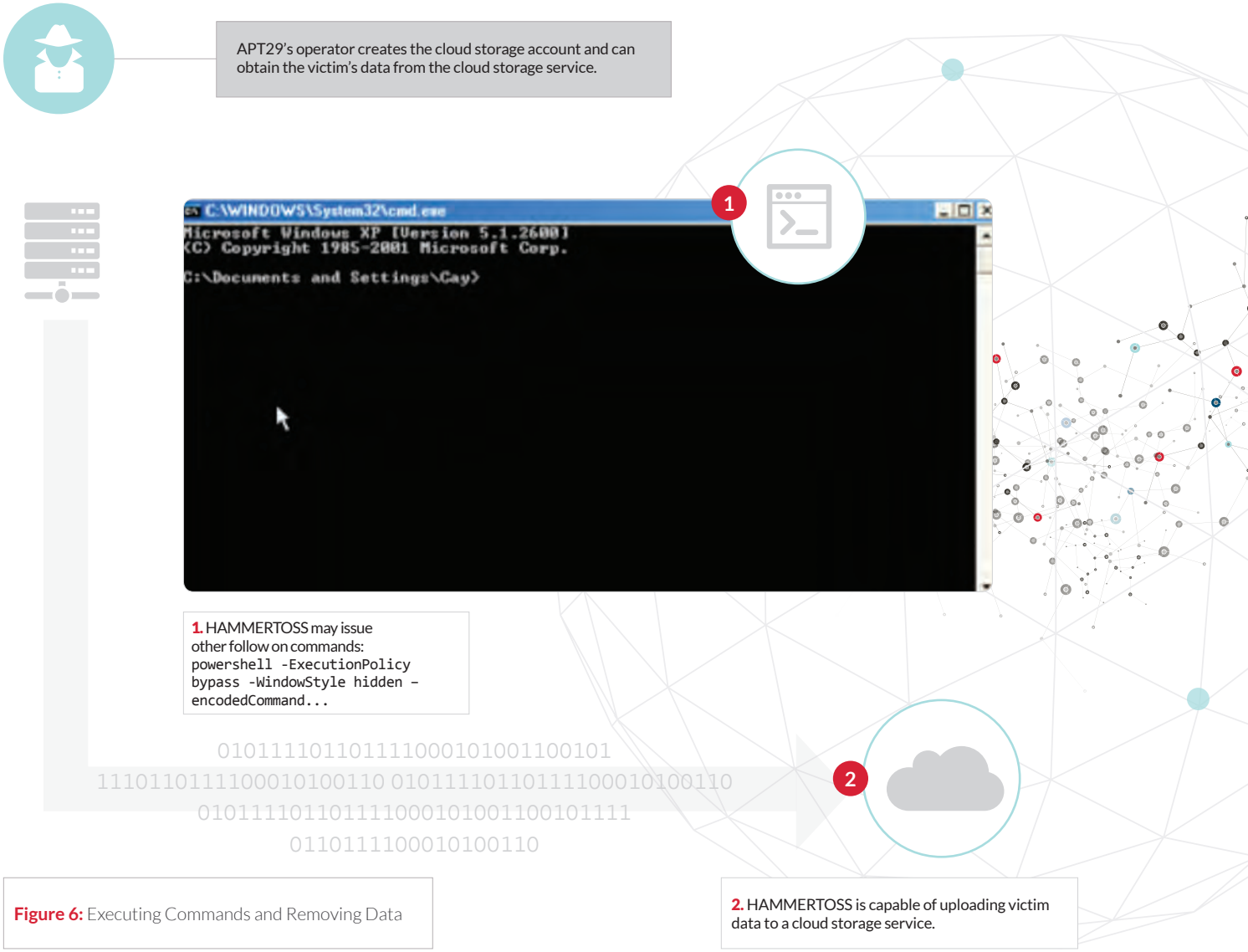


Figure 6: Executing Commands and Removing Data

The encrypted data in the image may include instructions to execute commands via PowerShell, execute a direct command or file, or save an executable to disk and execute it.

In several cases, the commands directed HAMMERTOSS to upload information from victim networks to accounts on cloud storage services using login credentials received in Stage 4. In our GitHub example, the decrypted data instructed the backdoor to obtain a list of running tasks—reconnaissance on the victim network—and upload it to a specific account on a cloud storage service using the login credentials. APT29 can then easily obtain the extracted information from the cloud storage service at their convenience.

CONCLUSION

Difficulty Identifying Accounts, Discerning Legitimate and Malicious Traffic, and Locating the Payload

HAMMERTOSS undermines network defenders' ability to identify Twitter accounts used for CnC, discern malicious network traffic from legitimate activity, and locate the malicious payloads downloaded by the malware.

- Identifying daily potential Twitter accounts requires network defenders to have access to the associated HAMMERTOSS binary and to reverse engineer it to identify the basename and the algorithm used to create the potential accounts. Monitoring malicious tweets from these accounts is difficult as each sample is capable of generating hundreds of potential Twitter accounts annually, and APT29 may only register a small number of those accounts for CnC.
- Employing legitimate web services that are widely allowed in organizations' networks—some of which use Secure Sockets Layer connections that ensure the communications are encrypted—makes it harder for network defenders to discern between malicious and legitimate traffic.
- Using steganography and varying the image size makes the target payload—the image containing the appended, encoded commands—less predictable. Even if the network defenders are able to predict or identify the target payloads, they need the associated HAMMERTOSS sample and relevant tweet containing the related encryption key information to decrypt the contents.

APT29: AN ADAPTIVE AND DISCIPLINED THREAT GROUP

HAMMERTOSS illustrates APT29's ability to adapt quickly during operations to avoid detection and removal. For example, if an organization blocks access to GitHub, APT29 could easily redirect HAMMERTOSS to download an image with encrypted instructions from another website. Similarly, if an organization starts monitoring Twitter activity on their network, APT29 could easily switch to using the *Uploader* variant of HAMMERTOSS, which does not use Twitter and communicates directly to a specified URL. If an organization identifies the handle generation algorithm and attempts to research old Twitter accounts, tweets, or secondary URLs, APT29 could easily delete previously used accounts or the locations where images were stored.

While each technique in HAMMERTOSS is not new, APT29 has combined them into a single piece of malware. Individually, each technique offers some degree of obfuscation for the threat group's activity. In combination, these techniques make it particularly hard to identify HAMMERTOSS or spot malicious network traffic; determine the nature and purpose of the binary; discern the malware's CnC method and predict its CnC accounts; capture and decode second-stage CnC information; and pinpoint and decrypt the image files containing malware commands. This makes HAMMERTOSS a powerful backdoor at the disposal of one of the most capable threat groups we have observed.

To download this or other
FireEye Threat Intelligence reports,
visit: https://www.fireeye.com/reports.html

FireEye, Inc. | 1440 McCarthy Blvd. Milpitas, CA 95035 | 408.321.6300 | 877.FIREEYE (347.3393) | info@fireeye.com | **www.fireeye.com**