C&C-AS-A-SERVICE: ABUSING THIRD-PARTY WEB SERVICES AS C&C CHANNELS

Artturi Lehtiö
F-Secure, Finland

Email artturi.lehtio@f-secure.com

ABSTRACT

A secure, reliable and undetectable method of communicating with and controlling malware is essential for modern malware operations. But designing, implementing and maintaining your own communication infrastructure isn't an easy task. Coincidentally, malware operators aren't the only ones interested in secure and reliable communication. Popular web services also want to provide their customers with a secure and reliable service. Add to that the fact that popular web services generate large amounts of indistinguishable web traffic to blend into and it starts to sound irresistible. Unsurprisingly then, recent years have seen a growing trend among malware operators of abusing third-party web services such as *Twitter*, *Facebook* and *Gmail* as command-and-control (C&C) channels.

This paper explores the multitude of ways in which modern malware abuses third-party web services as command-and-control channels. Through real life examples – from common cybercrime to targeted nation-state espionage – the paper provides a comprehensive overview both of the methods employed by malware and of the web services most commonly abused. The paper further analyses the benefits and disadvantages for malware operators when they use third-party web services as command-and-control channels. Finally, the paper also examines the challenges that such methods pose to the detection and prevention of malware.

SIMPLICITY FOR THE WIN: BACKDOOR.MAKADOCS

At its simplest, gaining some of the benefits of a third-party web service as a C&C channel doesn't require the complete redesign of a C&C protocol. A good example is Backdoor.Makadocs, which originally used HTTP to communicate with an attacker-operated web server. Later versions of the malware, however, do not connect directly to the malicious server. Instead, they route their traffic through *Google Docs* [1].

Google Docs has a little known feature called *Google Docs Viewer* that allows users to view documents or web pages from anywhere on the Internet via *Google Docs* (see Figure 1). Backdoor.Makadocs exploits this feature to essentially use *Google* as a proxy for its communications. Backdoor.Makadocs will connect to hxxps://docs.google.com/viewer?url=<actual C&C URL> where the URL of the actual Makadocs command-and-control server is passed as a parameter to *Google Docs Viewer*. *Google*'s service will then connect to the actual C&C URL, passing along any parameters specified by Makadocs, and display the C&C server's response back to Makadocs.

The end result of all this is increased anonymity, reliability and stealth for Makadocs' communications. Since the traffic to *Google* is over HTTPS, it's impossible for anyone monitoring traffic on the victim's machine or the victim network to distinguish traffic generated by Makadocs from traffic generated by the legitimate use of *Google Docs*. Therefore, it is impossible to detect Makadocs infections simply based on its communications with its C&C server. Likewise, since the URL of the actual C&C server will be encrypted in the traffic, identifying the C&C server from the traffic is similarly impossible. The only way to block traffic from Makadocs would be to block any traffic to *Google Docs*. However, this is unlikely to happen due to the popularity of *Google*'s services in legitimate use cases. The only exception to the above statements is a situation where the victim organization is actively filtering the contents of HTTPS traffic via man-in-the-middle techniques.

As can be seen, Makadocs uses a simple technique to abuse *Google Docs* and gain significant benefits for its C&C channel. The main downside for Makadocs is that *Google* will in all likelihood have logs of the traffic generated by Makadocs. Therefore, researchers or law enforcement working with *Google* might actually be able to gain better insight into Makadocs' historical activity. This, however, is only a small condolence for the victims and their organizations.

Figure 1: Screenshot of Google Docs Viewer rendering the Virus Bulletin homepage.

ABUSING THIRD-PARTY SERVICES FOR PRIMARY C&C CHANNEL ESTABLISHMENT

An apparently popular use case for abusing third-party services is for establishing a primary command-and-control channel. The essence of the method is usually the same: encoded messages containing the address of the actual C&C server are hosted on third-party web services where they are easy to retrieve, easy to update and difficult to block.

The report 'Shadows in the Cloud' documents a large cyber espionage operation that included the use of *Twitter*, *Google Groups*, *BlogSpot*, *Baidu Blogs* and *blog.com* to host messages containing the addresses of actual C&C servers [2]. Analysis of 'Operation Poisoned Hurricane' indicates that *Google Code* projects were used to host encoded addresses of C&C servers [3]. In more recent cases, attackers have taken to abusing the commenting functionality often present in web services. For example, APT17 embedded C&C addresses in comments on *Microsoft*'s *TechNet* website [4], and Janicab embedded C&C addresses in comments on *YouTube* (see example in Figure 2) [5]. The trojan downloader f0xy uses popular Russian social media site *VKontakte* for obtaining the address of its primary C&C server [6].

In all of the aforementioned examples, the primary command-and-control channel still links to an attacker-operated server via more traditional methods. The abuse of a third-party service therefore doesn't hinder network-based detections of malicious activity. Likewise, it doesn't significantly affect possible investigations of the malicious activity. The primary benefit for the attacker in this case is simply an easy and reliable way of broadcasting new C&C server addresses whenever the existing C&C servers are taken down or blocked.

Interestingly, such a method of operation on the attackers' part actually has benefits for defenders and researchers as well. Once attacker-controlled *Twitter* accounts, *Tumblr* accounts, or whatever the attackers are using, are identified, defenders and researchers can monitor them just as easily as the malware can. This way defenders can quickly react to attackers changing infrastructure and researchers can continue tracking the malicious activity. Unlike when operating their own servers with visitor logs, when the content being monitored is hosted by a third party, the attackers have no way of knowing if they are under surveillance.

Figure 2: Example of comments containing encoded C&C addresses for Janicab.

Additionally, the use of a public third-party service can sometimes make it easier for researchers to uncover additional malicious activity by the same attackers. In many of the cases mentioned in this section, the same markers that the malware used to identify messages intended for itself, were used by researchers to find otherwise unconnected user accounts, comments or blog posts with similar encoded messages and created by the same attackers.

Unless the attackers are vigilant, many services will also provide researchers with easy access to historical data such as old Tweets or old blog posts, thereby making it easier to track earlier activity by the same attackers. Other web services may

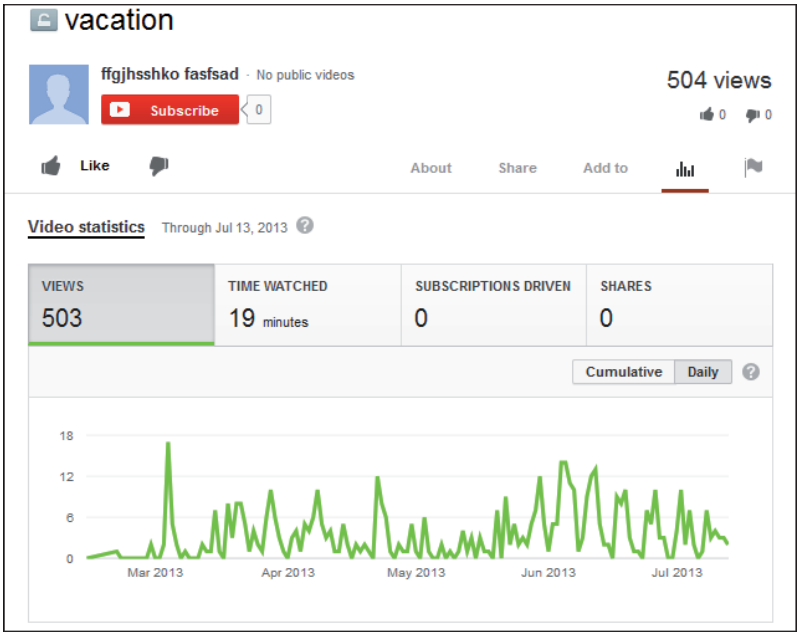

Figure 3: Daily view counts for a YouTube video, used by Janicab, showing historical activity [5].

display usage statistics, such as view counts, to users. In Janicab's case, researchers were able to make assumptions about Janicab's historical activity based on the daily view counts, seen in Figure 3, of the *YouTube* videos used by Janicab [5].

ABUSING THIRD-PARTY SERVICES FOR MALICIOUS DOWNLOADERS

In addition to abusing third-party services for broadcasting the addresses of primary C&C servers, similar methods are sometimes employed by malicious downloaders for broadcasting the locations of payloads to download. Infection with the alleged Russian espionage tool MiniDuke usually began with a small downloader that would search the Tweets of a specific *Twitter* user for encoded addresses of the next stage component to download and execute [7]. Similarly, the Svelta family of information-stealing malware includes a downloader component that will search pre-determined *Twitter*, *Jaiku* and *Tumblr* accounts for encoded links to the locations of additional components for downloading and executing [8].

The consequences of such techniques are, for both the attackers and defenders, the same as in the previous use case of establishing an actual command-and-control channel. In practice, both this and the previous use case are so similar that it should come as no surprise that some malware does both. For example, Trojan.Whitewell uses *Facebook* both for coordinating primary C&C channel establishment as well as for the downloading and executing of additional binaries [9].

It should be noted that using third-party services to host malware also appears to be popular among malicious actors. In such cases, the consequences for both attackers and defenders are similar to what has previously been discussed. However, solutions for hosting malware, while relevant to the workings of malicious downloaders, are not strictly related to solutions for malware command-and-control channels and are therefore beyond the scope of this paper.

THIRD-PARTY SERVICES AS BACKUP CHANNELS

In the previously discussed cases, third-party services have been abused in part to better facilitate situations where the primary command-and-control channel becomes unavailable. Closely related to those cases are ones where malware is designed to contact third-party services solely when the primary command-and-control channel is unavailable.

For example, OnionDuke samples [10] will often specify, in their configuration data, the name of a *Twitter* account. If OnionDuke is unable to contact the primary C&C server specified in its configuration, it will attempt to search for Tweets from the configured *Twitter* account, expecting them to contain links to image files embedded with updated versions of itself.

Malware authors sometimes employ a domain generation algorithm to combat the blocking or sinkholing of command-and-control server domain names. Similarly, OnionDuke also employs a sort of backup of a backup in the form of an algorithm that generates *Twitter* account names based on the current date. If OnionDuke is unable to find any Tweets from the *Twitter* user specified in its configuration, it will use the algorithm to generate a second username and check whether that user has Tweeted anything.

However, once a generation algorithm has been reverse-engineered, it is easy for defenders or researchers to register such accounts and thereby prevent the attackers from using them. Likewise, *Twitter* may be willing to help by taking down accounts associated with malicious activity. However, Flashback, a trojan targeting *Mac OS X*, has a solution to this problem. Flashback also uses *Twitter* and a generation algorithm as backup, but instead of searching for Tweets from a specific user, Flashback's algorithm will generate a hashtag and it will proceed to search for any Tweets containing that hashtag. This way, there is nothing for defenders or researchers to register and no single user account for *Twitter* to take down.

Again, the same advantages and disadvantages apply for both attackers and defenders as have applied in the previous use cases, although it is interesting to note that in OnionDuke's case, the backup method is not used to update the address of the primary C&C server but to download and execute a completely new piece of malware. In all of the observed cases, the malware was a newer version of OnionDuke, but the same mechanism could easily be used to download and execute a completely different malicious tool instead.

There is also another small but significant difference in the way OnionDuke abuses *Twitter* versus what the other previously mentioned examples do. In all the other cases, the C&C address or download location has been embedded in the Tweet, comment or blog post either inside a specific identifying string or encoded in a specific way, or sometimes both. Once again, this means that as soon as researchers know what to look for, it is easy for them to find additional suspicious content. OnionDuke, however, has moved the

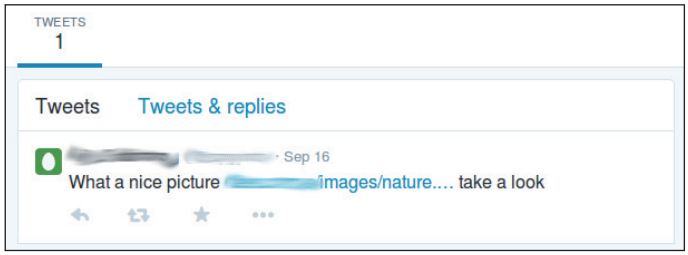

Figure 4: Example of a Tweet associated with OnionDuke and containing a link to an image file (shown in Figure 5) that embeds an updated version of OnionDuke.

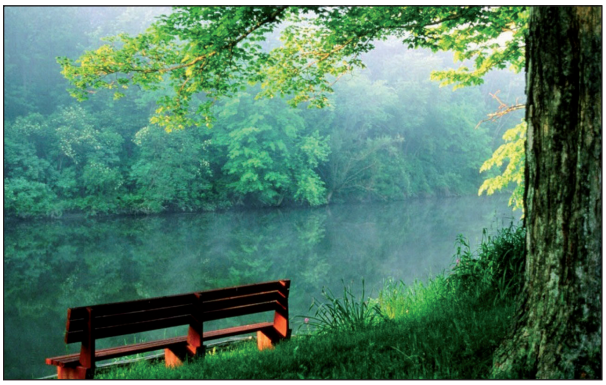

Figure 5: The image linked to by the Tweet in Figure 4. The image actually embeds an updated version of OnionDuke.

identifying string or signature from the content of the Tweet to the file downloaded from the location linked to in the Tweet. In all of the observed cases, the contents of the Tweet have appeared completely innocuous and human (an example is presented in Figure 4). Additionally, the links have pointed to legitimate image files that don't appear at all suspicious at first sight (an example is shown in Figure 5). This makes tracking down additional OnionDuke-related *Twitter* accounts or Tweets much more difficult.

THIRD-PARTY SERVICES AS PRIMARY CHANNELS

Previous examples have focused on abusing third-party services for specific use cases while still utilizing more traditional methods as the primary command-and-control channel. It is, however, entirely possible, and actually surprisingly common, for malware to rely solely on third-party services for all aspects of their command-and-control channel.

Social media services are popular not only among teenagers but also among malware authors. Slides leaked by whistleblower Edward Snowden detail an operation known as 'Byzantine Hades' that included malware which used *Facebook* as its command-and-control channel [11]. TwitterNET Builder provides a simple tool for creating malware that uses *Twitter* as its C&C channel [12]. Unlike teenagers, however, malware authors don't appear to be trend conscious when choosing which social media service to abuse. BlackEnergy2, for instance, has occasionally used *Google Plus* as a C&C channel [13].

In all of these cases, malware authors are exploiting the commonality of legitimate traffic to and from social media services. Due to this and the *de facto* use of HTTPS when communicating with social media services, it is easy for malware to hide its communications in the midst of legitimate traffic. Likewise, it's often a safe bet for malware authors to assume that traffic to social media services will not be blocked, ensuring added reliability for the C&C channel.

Similar to social media services, webmail services are widely used services that provide malware authors with an easy way to implement a stealthy and reliable command-and-control channel. For example, IcoScript would log into a predetermined *Yahoo Mail* account, compose a new email containing data to exfiltrate and then search for existing emails containing new commands to execute [14]. Some of the malware documented in the report 'Shadows in the Cloud' used *Yahoo Mail* for command and control as well as data exfiltration. Trojan.Gmail used similar techniques for *Gmail* instead of *Yahoo Mail* [15]. Again, attackers are able to avoid both network-based detections and the blocking of their malicious communications by only generating network traffic that is indistinguishable from legitimate traffic.

Some malware intends to provide attackers with interactive remote access to victim computers in much the same way as SSH or Telnet might be used. In such cases, social media or webmail services may not be the best choice. Instant messaging services, on the other hand, are a perfect match as they are specifically designed for interactive communication between two distant parties. Multiple malware families associated with the group APT1 did just this. GLOOXMAIL used *Google Talk* and MACROMAIL used *MSN Messenger* to ferry commands and results between attackers and victims [16].

Again, such services are also commonly used for legitimate purposes and therefore the traffic is unlikely to raise suspicion or get blocked. Many corporations even use services such as *Microsoft*'s *Skype* for official internal communication. The only real downside for the attackers is when abusing a service that doesn't inherently provide end-to-end encryption. In such cases, the service provider may later be able to retrieve the contents of the malicious communications and provide it to law enforcement or researchers. In APT1's case, the malware authors took this possibility into account by implementing their own encryption on top of the instant messaging service they were using [16].

The potential for abusing social media, webmail or instant messaging services may seem obvious. In some cases, however, malware authors have been more creative. For instance, APT1's toolkit also included the trojan CALENDAR which used *Google Calendar* as its C&C channel by storing commands and responses as calendar events.

Mobile also brings new opportunities for malware authors. Mobile applications often use push notification services to provide parts of their functionality. Unsurprisingly, malware authors have also taken to using these services as command-and-control channels. The Cajino trojan, which targets *Android* devices, used the push notification service *Baidu Cloud Push* as its C&C channel [17]. Similarly, multiple trojans targeting *Android* devices have been observed abusing *Google*'s *Cloud Messaging* push notification service as a C&C channel [18].

In all of the mentioned examples, by basing the entire command-and-control channel on a third-party service, attackers are able to gain significant advantages. The most impacting of these are probably the advantages gained in stealth and reliability as communication happening solely via third-party services is very difficult to detect or block. However, other, possibly not so obvious benefits may also be available. For instance, why go to the trouble of implementing your own protocol supporting real-time interactive remote control of victim computers when instant messaging service providers struggle with many of the same challenges but probably with larger development budgets and greater resources?

THIRD-PARTY SERVICES FOR EXFILTRATING STOLEN DATA

There remains one more use case to be singled out in this paper, which is the abuse of third-party services for exfiltrating stolen data. The transfer of large numbers of files out of an organization's computer network is often cause for suspicion. However, popular file-sharing or cloud storage services, such as *DropBox* or *Microsoft OneDrive*, are specifically intended for transferring large numbers of files to and from computers. Therefore, by abusing such a service for the exfiltration of stolen data, attackers may be able to avoid raising suspicions and having their activity detected.

For example, attackers using the espionage tool CozyDuke would often exfiltrate stolen data from victim networks via *Microsoft OneDrive* accounts [19]. In other cases, the apparent emphasis on data exfiltration has been so great that the entire command-and-control channel has been based on a

cloud storage service. An example is the Inception Framework, which used the cloud storage service *CloudMe.com* as its C&C channel [20]. An interesting example of the opportunities provided by abusing cloud storage services is also the DropSmack research performed by Jake Williams [21], which details how he used *DropBox* during a penetration testing assignment for everything from infection vector and command-and-control channel to data exfiltration channel.

Once again, by exploiting the popularity and legitimate use cases of a third-party service, attackers are able to better avoid detection and increase the reliability of their operations.

IMPLICATIONS FOR DEFENDERS, RESEARCHERS AND LAW ENFORCEMENT

By far the greatest implications of malware abusing third-party services as command-and-control channels are for the network-level detection and prevention of malware. Third-party web services commonly support HTTPS. Malware abusing third-party web services commonly take advantage of this to encrypt the contents of their communications. Unless organizations actively engage in intercepting HTTPS traffic (with its own set of issues), most such malware-generated traffic to third-party web services is indistinguishable from legitimate traffic. Therefore, network-level detection is impossible. Likewise, network-level prevention would require an organization to block all traffic – both legitimate and malicious – to third-party services which, owing to the popularity of many such services, is unlikely to happen in most organizations. Heuristic or similar methods may enjoy some success in identifying malicious traffic to third-party services. However, this is an area that clearly would benefit from more research.

Slightly easier to detect and prevent are cases of malware using protocols other than HTTPS to communicate with third-party services. For example, an organization may decide to ban the use of third-party services such as cloud storage or instant messaging services. In that case, any traffic associated with such services could be seen as suspicious and blocked by default. Once again, however, the issue is that many organizations rely on such services for legitimate use cases. *Skype* (formerly *Lync*), for instance, is popular for intra-organizational communication.

The abuse of third-party services by malware also has wide-ranging implications for investigating malicious activity. If the only command-and-control channel is via a third-party service, there are no malicious domain names or IP addresses for researchers or analysts to track. Likewise, the technique of sinkholing C&C server domain names can't be used if the only domain names used by the malware are associated with legitimate third-party services. Exceptions, however, are cases such as OnionDuke, which uses an algorithm to generate *Twitter* usernames. With the service provider's co-operation, 'sinkholing' – registering the usernames before attackers do so – would be possible and would, for instance, enable law enforcement to take control of such a botnet.

Another implication for investigations, caused in part by the previously discussed difficulty of identifying malicious traffic to third-party services, is the difficulty of using network traffic logs, netflow data and similar to help investigate an attack. Traditionally, traffic logs can help investigators identify further infections or trace the attacker's activity over time. If the malicious traffic can't be distinguished from legitimate traffic, this method can't be used.

Similarly, obtaining netflow data or packet captures of traffic to and from a command-and-control server is often helpful in understanding the scope of a malware operation and in tracking down the attackers. This, naturally, is not possible if there is no attacker-controlled server to investigate. In this case, however, attackers abusing third-party services for C&C channels may actually be beneficial for investigators. It's probably safe to assume that all third-party service providers store at least some level of logs of the use of their services. With the co-operation of service providers, investigators may be able to use these logs to track both victims and attackers.

Sometimes investigators or researchers are even able to obtain historical data on a malware operation thanks to the abused service storing such information. For example, while investigating Trojan.Grups, which used *Google Groups* as a C&C channel, researchers were able to trace the historical activities of the attackers because *Google Groups* is designed not only to show all messages ever posted to a newsgroup but also what modifications may have been made to those messages [22]. Similarly, while investigating the Inception Framework, researchers were able to work together with the service provider both to discover additional malicious use of the service as well as obtain archived copies of data that the attackers had deleted from the service [20].

Even in situations where usage history or other relevant data is not publicly available to users of a service, it is probably safe to assume that co-operation between investigators and the third-party service providers in order to obtain such data is likely to be easier and more fruitful than, for instance, attempting to obtain similar data from bulletproof hosting providers working hand in hand with the attackers. In such cases, the insistence of the attackers on abusing third-party services is actually beneficial for researchers and law enforcement.

Another situation where researchers may actually benefit from attackers abusing third-party services is when attempting to monitor malicious activity. A researcher can, for example, easily monitor the *Twitter* feed of a known attacker-controlled *Twitter* account for any new command-and-control server addresses, download locations or commands. Were the attackers operating their own malicious server, they might be able to detect researchers monitoring the server via logs, or they might attempt to block access to the server altogether for researchers with methods such as IP blacklisting, country-based restrictions etc. In the case of a third-party service, however, such actions are not possible for the attackers. They therefore have no way of knowing when they are under surveillance.

The abuse of third-party services as command-and-control channels also has implications for how botnet takedowns are performed. Traditionally, such a takedown would include coordinating with registrars or hosting providers either to redirect malicious domain names away from attacker-controlled servers or to take control of the actual servers. In the case of abused third-party services, the equivalent would usually be to coordinate with the service provider to take control of user accounts associated with the malicious activity or to remove the malicious content from the services. In this regard, it's not so much a question of whether takedowns are

easier or harder to perform when third-party services are abused as it is simply a question of requiring a slightly different approach.

The real hindrance for defenders, researchers and law enforcement coordinating takedowns comes in the form of effort expended. The effort, time and resources required for an attacker to register a new user account and post new malicious content is negligible compared with the time and coordination required for a takedown. Of course, the same imbalance also applies when more traditional command-and-control channels are used. However, setting up a more traditional C&C channel usually entails more than the couple of mouse clicks in a web browser that it does to, for instance, Tweet a new message with a specific hashtag, as the operators of Flashback might do.

CONCLUSION

The idea of abusing third-party services as command-and-control channels is not new. It's also not as rare or as difficult as one might initially imagine. It is, however, in the author's opinion, an under-researched subject with wide-ranging implications for anyone working against malware authors and operators. It therefore deserves more attention.

The possibility of malware authors and operators increasingly abusing third-party services for malicious purposes poses serious challenges for the ways in which network defenders, malware researchers, analysts and investigators are used to working. You track malicious actors based on WHOIS information? That won't work. You rely on network-based detections for emerging threats and previously unseen malware? That won't work. You rely on network logs to triage and investigate compromises of your organizations computer systems? That won't work.

Attackers abusing third-party services force us to reorient. Some familiar methods stop working. But other, previously unavailable methods arise. Understanding the ways in which third-party services can be abused and the implications such actions can have for our work is crucial. We often complain that attackers seem to be one step ahead of us. Let's not let them get ahead by another!

REFERENCES

[1] http://www.symantec.com/connect/blogs/malware-targeting-windows-8-uses-google-docs.

[2] http://shadows-in-the-cloud.net/.

[3] https://www.fireeye.com/blog/threat-research/2014/08/operation-poisoned-hurricane.html.

[4] https://www2.fireeye.com/rs/fireye/images/APT17_Report.pdf.

[5] https://www.f-secure.com/weblog/archives/00002576.html.

[6] http://community.websense.com/blogs/securitylabs/archive/2015/01/30/new-f0xy-malware-employs-cunning-stealth-amp-trickery.aspx.

[7] https://securelist.com/blog/incidents/31112/the-miniduke-mystery-pdf-0-day-government-spy-assembler-0x29a-micro-backdoor/.

[8] http://asert.arbornetworks.com/twitter-based-botnet-command-channel/.

[9] http://www.symantec.com/connect/blogs/trojanwhitewell-what-s-your-bot-facebook-status-today.

[10] https://www.f-secure.com/weblog/archives/00002764.html.

[11] https://www.eff.org/files/2015/02/03/20150117-spiegel-byzantine_hades_-_nsa_research_on_targets_of_chinese_network_exploitation_tools.pdf.

[12] http://www.hotforsecurity.com/blog/twitter-controlled-botnet-sdk-at-large-813.html.

[13] https://securelist.com/blog/68838/be2-extraordinary-plugins-siemens-targeting-dev-fails/.

[14] https://www.virusbtn.com/pdf/magazine/2014/vb201408-IcoScript.pdf.

[15] http://www.nartv.org/2010/10/22/command-and-control-in-the-cloud/.

[16] http://www.mandiant.com/apt1.

[17] http://b0n1.blogspot.fi/2015/03/remote-administration-trojan-using.html.

[18] https://securelist.com/blog/mobile/57471/gcm-in-malicious-attachments/.

[19] https://www.f-secure.com/documents/996508/1030745/CozyDuke.

[20] http://dc.bluecoat.com/Inception_Framework.

[21] https://media.blackhat.com/eu-13/briefings/Williams/bh-eu-13-dropsmack-jwilliams-wp.pdf.

[22] http://www.symantec.com/connect/blogs/google-groups-trojan.