HOW THEY'RE GETTING THE DATA OUT OF YOUR NETWORK: A SURVEY OF METHODS USED FOR EXFILTRATION OF SENSITIVE DATA, RECOMMENDATIONS FOR DETECTION AND PROTECTION

Eric Koeppen
IBM, USA

Email erkoeppe@us.ibm.com

ABSTRACT

When malware infects a system, often it is only the first step in a chain of events. Once on a system, malware can move laterally through a network, infecting other systems, and searching for important data. If the malware finds data for which it has been programmed to search, or an attacker is using the malware to poke around opportunistically, it can then send copies of that data to external servers in a process known as exfiltration.

We have seen exfiltration used in many attacks where confidential customer information has been leaked to malicious actors. Such infections can have disastrous effects on a company's brand, customer loyalty, and competitive advantage.

Starting in 2006, Operation ShadyRAT targeted 72 different companies over a period of five years, exfiltrating massive amounts of information. In the famous *Target* breach of 2013, 40 million credit and debit card accounts were stolen, along with PII data on another 70 million customers. *Target* said its data breach has cost $240 million so far, with further litigation threatening to push that cost still higher.

This paper will examine some examples of malicious data exfiltration and explore methods for detecting and mitigating against such threats.

INTRODUCTION

As malware authors become more creative, and digital defences become more complex, there are bound to be some casualties of the information security arms race. No longer content with the thrill of wreaking havoc on a system, today's hackers want to see a return on the investment of time and energy they put into infiltrating systems. Gaining a foothold in a system is only the first step in a chain of events that constitute a modern compromise. Once malware has established itself on a system, it can move laterally throughout a network, infecting system after system, and searching for important data which can then be exfiltrated (i.e. sent to external servers).

Exfiltration has been used in many recent attacks where sensitive information has been leaked to malicious actors. This sensitive information may be proprietary data that drives a company's business, or private customer data, including financial information. Such infections can have disastrous effects on a company's brand, customer loyalty, and competitive advantage. Not only can a company lose money through the direct loss of revenue, but the leaking of customer information can harm consumer confidence in the company in the long term, it can lead to costly litigation, and the leaking of trade secrets can compromise the company's position in the marketplace.

In the rest of this paper, we will look at some examples of malware performing data exfiltration, discuss methods for identifying these types of data breaches, and explore methods for mitigation of data exfiltration.

EXFILTRATION SCENARIOS

Advanced persistent threat (ShadyRAT) exfiltration

Starting in 2006, Operation ShadyRAT targeted 72 different companies over a period of five years, exfiltrating massive amounts of information. When it was reported by *McAfee* [1], it was one of the largest advanced persistent threat (APT) campaigns ever made public. ShadyRAT leveraged various infection mechanisms to gain footholds in affected systems, often using targeted spear phishing emails to trick users into opening infected attachments. Once on a system, the malware connected to a remote server for command and control (C&C) instructions and exfiltration. *Dell* researcher Joe Stewart linked Operation ShadyRAT definitively to the Chinese threat group Comment Crew [2] (a.k.a. APT1 [3]).

ShadyRAT was particularly novel in the way it dealt with C&C communications, often hiding commands in images via steganography [4] and mathematically encoding commands into the image data. One of the C&C commands the remote server can issue on a compromised system is the instruction to connect to another remote server on a specified port. When this connection is made, a malicious user on the remote server to which the infected system connects is given a small command-line shell to navigate through the infected system. This shell allows the remote user to browse through the infected system, looking for any interesting data, which it can then transmit to the remote server. Unfortunately, the format of the data being exfiltrated, and even what data is considered 'interesting' for exfiltration, is nearly impossible to predict. Filtering for data leaving the system must thus be based on system-specific guidelines. The initial connection between the infected system and the remote controller involves a predefined handshake:

```
/*\n@***@*@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@>>>>\*\n\r"
```

It should be possible for most intrusion prevention systems (IPS) to filter on this string. However, if the string is prone to false positives, then another strategy, such as the use of access control lists for the machines that contain sensitive information, would be preferable.

Point of sale malware exfiltration

In the *Target* case that came to light in 2013, 40 million credit and debit card accounts were stolen [5], along with PII data on another 70 million customers. *Target* has reported that the data

breach has cost the company $240 million so far [6], with further litigation threatening to push that figure higher. The *Target* attack is representative of point of sale (POS) malware. POS malware typically infects a system via a trojan. Once it has gained a foothold in the system, POS malware typically performs memory scraping on all running processes in search of credit card track data. Credit card track 1 and 2 data (separated by a ';') appears in the following format:

```
%B1234567890987654^LastName/
FirstName^1407777000000001000000003000000?; 1234567890
987654=1407777000000300001?
```

This breaks down as follows:

1234567890987654	16-digit credit card number
LastName/FirstName	First and last name of card holder
1407	Expiration date in the format YY/MM (year/month)
777	Service code
000000001000000003000000	Track 1 discretionary data (may include CVC1)
000000300001	Track 2 discretionary data (may include CVC1)

When POS malware finds data formatted like this (or a subset of the above) in process memory, it sends it through one of various transport protocols to a remote server. Occasionally, POS malware will perform a checksum on suspected credit card data using the Luhn algorithm [7] to determine whether it is a valid credit card [8].

Often, the data is reformatted for transmission. Commonly, track 2 data (i.e. the data following the ';' in the example above) will be stripped of the '=' sign and sent out. Certain variants of POS malware do encrypt the data for transmission, which makes credit card data exfiltration very difficult to detect via content filtering. The transport methods used by a subset of POS malware observed in the past year have included HTTP Posts, HTTP Gets, FTP and SMB/NetBIOS. The *Target* POS malware connects to a specific UNC path and copies data to a shared directory [9]. IPS signatures based on searching for valid credit card numbers will often catch the credit card data being exfiltrated as long as the data is not encrypted. If a local intrusion detection mechanism can detect attempted reads across process memory, this will help with identifying memory scraping. If memory protection can be put in place that prevents access to memory across the various processes, this should prevent the memory scraping, which would prevent the credit card data being gathered in the first place.

Financial malware exfiltration

Variants of the Zeus banking trojan have terrorized the banking industry for years, accounting for losses of hundreds of millions of dollars. The Gameover Zeus variant alone has accounted for over $100 million in theft since 2011 [10]. There are many different variants of the Zeus banking trojan and they use various methods for exfiltrating banking information.

Variants of the Zeus trojan have been known to check the *Internet Explorer* browsing history and address bar for a list of known banking websites. If any such sites are found, Zeus prepares a mock-up of the site which is then served to the victim when they try to browse to it. The mocked up web pages are actually just a ruse to harvest login credentials for those sites. Once gathered, the login credentials are encrypted, stored in a data file, and then transmitted to a remote server via HTTP Post [11]. In this case, by the time the data is exfiltrated, it's very late in the process. Stopping the infection before it gets to that point is no doubt preferable. In order to catch the outbound traffic, filtering on known servers (typically serving 'coolstar.php') can be very helpful.

Beyond just checking for web history, Zeus variants have been known to ransack systems for other sensitive data. The search for sensitive data either to steal or to make additional use of includes the following:

- Parsing cookie files for local data-containing files
- Stealing digital certificates
- Stealing local private keys
- Stealing FTP client information
- Parsing registry keys for valuable information
- Stealing settings for mail clients like *Windows Live Mail* and *Outlook*.

Once the above data has been harvested, it is saved in a file which is then encrypted and sent back to a remote server. As above, it is important to monitor the system locally and catch the malware before the encryption occurs.

Zeus variants have been very successful with keylogging methods and have even created graphical keyloggers that capture screenshots in the case of online banking sites that use graphical or virtual keyboards instead of traditional keyboard-based input methods. Often this data is encrypted before exfiltration, so network-based detection of the data itself becomes extremely difficult. Use of process monitoring and memory protection to detect and stop the malicious processes that gather the information becomes key. Since content filtering on encrypted data is a serious issue, it is all the more important to specify and lock down the servers with which machines containing sensitive data are allowed to communicate. In such a case, it is important to have a blacklist of known exfiltration servers.

Recently, Zeus variants have started to use P2P proxy bots [12]. These proxy bots gather stolen data from other bots in the botnet for exfiltration. Payload messages are all hashed, signed, and then encrypted with RC4 encryption. Detecting the presence of the P2P botnet on the network is key, and can be done by tuning network monitors to detect the P2P keep-alive messages. Keep-alive messages typically have the fourth byte set to 0x00 as a version request type [12]. If a peer fails to answer version request type messages within five tries, then the malware will try to connect to www.google.com or

www.bing.com to verify that it has Internet access. Detecting these messages in amongst legitimate HTTP traffic is not always feasible.

CONCLUSION

There is an arms race going on between those who wish to protect sensitive data and those who wish to gain unfettered access to it for their own purposes. Malware authors are becoming extremely creative, not only with their infection methods, but also with the methods they use for the exfiltration of sensitive data. This paper has described some widely used data exfiltration scenarios, and has suggested methods for detection and protection where possible. We saw with advanced persistent threat (ShadyRAT) exfiltration, that outbound traffic between the infected host and the remote server can be identified by a predefined handshake token. With the point of sale malware exfiltration, we saw that data can be filtered on if it is not encrypted, and if it is similar to known data formats for credit card track 1 and track 2 data. With financial malware exfiltration, we saw that local detection/prevention is key, as the data is typically encrypted before exfiltration. However, we did also see that internal proxy keep-alive communications can be detected for peer-to-peer implementations. This paper should help enhance the arsenal for those looking to detect and protect data that is leaving a compromised network.

REFERENCES

[1] http://www.mcafee.com/us/resources/white-papers/wp-operation-shady-rat.pdf.

[2] https://www.ncsc.nl/binaries/nl/conference/conference-2011/speakers/joe-stewart/1/Presentation%2BJoe%2BStewart.pdf.

[3] http://intelreport.mandiant.com/Mandiant_APT1_Report.pdf.

[4] http://www.symantec.com/connect/blogs/truth-behind-shady-rat.

[5] http://pressroom.target.com/news/target-confirms-unauthorized-access-to-payment-card-data-in-u-s-stores.

[6] http://finance.yahoo.com/news/target-data-breach-cost-banks-140004350.html.

[7] http://www.google.com/patents/US2950048.

[8] http://securityaffairs.co/wordpress/25479/cyber-crime/soraya-pos-malware.html.

[9] http://h30499.www3.hp.com/t5/HP-Security-Research-Blog/An-evolution-of-BlackPOS-malware/ba-p/6359149#.U5u0iVZdXXA.

[10] http://www.bloomberg.com/news/2014-06-02/russian-charged-with-running-100-million-data-theft-plot.html.

[11] https://kc.mcafee.com/resources/sites/MCAFEE/content/live/PRODUCT_DOCUMENTATION/23000/PD23030/en_US/McAfee_Labs_Threat_Advisory_PWS-ZBot.pdf.

[12] http://www.syssec-project.eu/m/page-media/3/zeus_malware13.pdf.

[13] http://www.tech-faq.com/layout-of-data-on-magnetic-stripe-cards.html.

[14] https://www.appsecconsulting.com/appsec-blog/searching-for-credit-card-track-data-in-memory/menu-id-193.html.

[15] http://krebsonsecurity.com/2014/01/a-first-look-at-the-target-intrusion-malware/.

[16] http://www.darkreading.com/vulnerabilities---threats/zeus-gameover-trojan-expands-global-reach-/d/d-id/1252756.

[17] https://labs.snort.org/papers/zeus.html.

[18] http://about-threats.trendmicro.com/us/malware/TSPY_ZBOT.XXT.