Malware Detection based on HTTPS Characteristic via Machine Learning

Paul Calderon[1], Hirokazu Hasegawa[2], Yukiko Yamaguchi[2] and Hajime Shimada[2]
[1]*Ensimag, Grenoble INP, France*
[2]*Information Technology Center, Nagoya University, Japan*

Keywords: Security, Malware, Infection Detection, Machine Learning.

Abstract: One of the major threat in today world are malwares that can infect computers. In order to prevent infection antimalwares softwares are installed but if the malware it is not detected at the installation it will probably never be detected. Behavioural analysis is necessary. Most of nowadays malwares connect to C&C servers by utilizing HTTP or HTTPS in order to receive orders. In this paper a method of behavioural analysis focus on the observation on HTTP and HTTPS network packets will be presented. This analysis is made by using machine learning. We evaluated our method by using 10-fold cross validations. The experimental result shows that precisions and recalls are more than 96% in average.

1 INTRODUCTION

Executing malicious programs on a computer is one of the major threat and attack observed against informatics systems. In order to respond to this attacks, it become common to users to install antimalware softwares. Different methods are used by this softwares to detect malicious programs, for example static analysis based on the signature of files or dynamic analysis using ”sandbox” that try to detect suspicious behaviour. One way usually used to avoid static analysis detection is the encryption of the program. However this is usually not useful against dynamic analysis. Techniques exist to trick behavioural analysis, some of this techniques are relatively easy to use. All of this detection methods are made to prevent machine infection. Indeed early detection is very important, if a malware is detected after the infection, it could have already achieve is goal such as stealing data, mining bitcoins, etc. However it is not always possible to detect malwares in advance, some of them can avoid detection techniques. The malware could have been installed because of human error, or because the malicious behaviour is delayed. For this reasons it is also important to detect malwares once the machines are infected. This can be done by analysing permanently the behaviour of program used on computer. But it uses resources from the computer. Most of virus are now communicating on the Internet with the attacker through a C&C server. Analysing traffic network can be an option for discovering suspicious behaviour and detect infected machines. For example using a firewall blocking some protocols can be a possibility but most of malwares communicate through HTTP or HTTPS that are daily used while surfing on the web. It is then important to detect patterns on HTTP and HTTPS packets that allow to discriminate infected machines network packets from legitimate packets. Finding the good pattern can be tough, we are going to use machine learning to automatize the network analysis. One article presented at the ICOIN 2017(Ogawa et al.,) shows that analysing HTTP traffic is powerful method on detecting malware infected machine. As being able to detect malware infected machine with a precision and recall superior of 96%.

2 MALWARE'S COMMUNICATION

The first malwares were only doing harm to infected computers, without the ability to change their behaviour depending on what the attacker wanted. But having the possibility to control the malware at distance increase the power of threat of malwares, with the possibility to give them orders that modify their behaviour depending on the situation. In order to give order to malwares hackers use what it is called com-

Calderon, P., Hasegawa, H., Yamaguchi, Y. and Shimada, H.
Malware Detection based on HTTPS Characteristic via Machine Learning.
DOI: 10.5220/0006654604100417
In *Proceedings of the 4th International Conference on Information Systems Security and Privacy (ICISSP 2018)*, pages 410-417
ISBN: 978-989-758-282-0

mand and control (C&C) servers. Malwares communicate through internet with a remote server that give order to one or all the malwares that have infected computers. C&C is especially known with botnets. Botnets are computers that have been infected in order to be controlled remotely. Generally botnet do not do any harm to the infected computer except than sending unwanted internet traffic. Botnet are particularly known for sending spam, creating DDos attack, etc. Which in fact it can be harmful for the network and also for the user as the IP address can be blacklisted if detected for this kind of attack [1].

There are two main reasons for malwares to have the possibility to communicate.

First of all, the connection to a C&C server is made to modify the behaviour of the malware for exemple deciding when to launch an attack, or which data should be stolen.

Sending the stolen data is not the only reason for malwares to communicate. As malwares become more and more efficient, they tend to behave as legitimate programs. They can now install updates of their codes automatically[2]. Updating the malware's code is the second reason of enabling malware's communication. Being able to update a malware give the possibility to add new behaviour to a malware, new type of attacks but not only, this also give the possibility to avoid their detection. For exemple if a malware is discovered and the antimalwares softwares start to use the signature of the malware to detect it, it will become useless. By updating the malware it is possible to modify its signature and make it difficult the detection of already installed ones.

Now malwares try more and more to imitate normal traffic that are used by everyone.

For example some botnets use Twitter[3], Reddit or other well known web application to communicate with the hacker.

Malware programmers can also create their own custom C&C server that do not use irc or the recent social media.

3 RELATED WORKS

Analysing HTTP packets for security purpose was especially done for making IDS system. Using machine learning and the use of good features and especially algorithm is really important for creating good detectors(Maloof,). Often no so many features are necessary. Previous works (Ichino et al., 2015)(Otsuki et al., 2014) also show that HTTP headers contain a lot of information that can help for their detection. This methods were base on time slot information and the open ports for the applications. Otsuki et al. (Otsuki et al., 2014) made his work especially on detecting infection based on HTTP packets. In this paper it was proven that is possible to increase malware detection, for worms and trojan malware types, by using some of ASCII opcodes present in HTTP packets. It also show that worms and trojan use some HTTP client header far less than normal traffic and that give an effective way to help discriminating this kind of malwares from normal traffic. This method was also link with other features such as the average time slot interval or HTTP packet length. All of this method are made to avoid at maximum the content of HTTP packet, for privacy problems but also because this content can be encrypted which would not give enough, or accurate information. However this methods are only effective with HTTP which can be problematic with the increase of malwares using encrypted communication especially with HTTPS.

The work presented in this paper is a continuation of a paper : "Malware Originated HTTP Traffic Detection" (Ogawa et al.,). This paper presents how it is possible to detect infected computers by analysing HTTP traffic network. This is possible because now most of malwares communicate with the hacker through C&C servers. By the help of machine learning it was proven that it is possible to detect infected computers by analysis the HTTP network traffic generated by a computer. This is done by looking to the data of HTTP packet at the exception of body content, allowing kind of respect of privacy[4]. In the previous HTTPS network traffic generated by malwares was not studied, which will be done in this paper. The previous analyser was based on the following features : HTTP request interval, body size, and bag-of-words of the HTTP headers. The appearance ratio of host pairs was then calculated before starting the classification learning. The method was evaluated by 5-fold cross validation and gave an average value for the recall of 96%. The following array show a summary of the results. The following notation will be used : P_{Normal} for the precision in detecting normal traffic and $P_{Infected}$ for the precision in detecting infected traffic. The same notation is used for the recall by using R instead of P.

The protocol proposed in this paper is similar to the one used in the previous one. The protocol will be

[1] That would not be very nice not to be able to connect to Google because your computer tried to DDos it.

[2] Which is now done by automatically by most popular OS.

[3] https://www.welivesecurity.com/2016/08/24/first-twitter-controlled-android-botnet-discovered/

[4] Except that there is no privacy when using HTTP

Table 1: Average for k-means with k=100.

	P_N	P_I	R_N	R_I
Average	0.95	0.85	0.94	0.82

more detailed in the following section.
All this works show one common point which is that malware detection can be done only by analysing HTTP header, and some other features. This make an easy way to implement the detection as it can be made at the firewall level. However, it is only based on HTTP and HTTPS was not studied. In this paper will focus on HTTP and especially on HTTPS.
In June 23, 2017, Cisco published a paper about similar topic : "Detecting Encrypted Malware Traffic (Without Decryption)"(Cis,). The technique used different features from the one used in this paper. The features used by Cisco, as described in their paper, are the following ones :

- **Legacy**, it correspond to classic data used to analyse network traffic packets. They used the duration of the communication, the number of packets and number of bytes send by the client and send by the server. It was collected by the Cisco NetFlow.
- **Sequence of packet length** is also a classic feature as it correspond to the size of the packets sent and received.
- **TLS Metadata** is a framework that allow to retrieve information on TLS packets. It was used to fingerprint the client during the checkhand protocol. It allowed them to determine the origin of the certificate used for the exchange.

By using this features and a Random tree forest algorithm, Cisco was able to have 99.99% of the normal traffic correctly classified and 85.90% [5] for malware traffic. The results of this paper will be compared to ours in the result section.

4 TRAFFIC DETECTION METHOD

Most of the protocol used was mostly inspired from the method used in the previous paper. In this section will be presented how the data is prepared from raw pcap files, the way of analysing it and the testing method used to validate the protocol.

[5] precision and recall are not precised

4.1 Testing Method

Testing a protocol is the only way to determine its efficiency. In this section the testing method used to validate our work is detailed.

4.1.1 Machine Learning Implementation

In order to avoid bugs or problem with machine learning implementation, well known machine learning libraries were used. In this project, the Weka[6] framework has been used. This framework developed in Java by the University of Waikato posses a lot of machine learning algorithm already implemented. It also has a graphical interface that allows to observe the data very easily. For this reason this framework was used for testing our detection method. It also allows to use most of machine learning on data, by putting the data into an appropriate format through a file.

4.1.2 Datasets

Two sources of data were used in order to decrease the risk to use biased information. This two sources provided pcap files of malware network traffic or normal network traffic. The first data source is some data provided by NTT[7] a Japanese telecommunication company. The second data source are pcap files generated by the Georgia Institute of Technology[8] using Panda technology to retrieve it [9]. For the normal traffic network, it correspond to 12h of capture of network traffic generated by our lab.

4.1.3 Efficiency Measurement

The values used to determine the pertinence of a model are recall and precision as described in the machine learning section. This values have been measured through the weka framework. On every dataset k-fold cross-validation have been used with k equal 5 and 10 to verify that the results are the same.

4.2 Proposed Method

In this section will be explained the way of preparing the data for the machine learning algorithm step. The method is divided as follow:

1. Data retrieving

[6] http://www.cs.waikato.ac.nz/ml/weka/
[7] Nippon Telegraph and Telephone
[8] panda.gtisc.gatech.edu/malrec
[9] https://github.com/moyix/panda-malrec

2. Network traffic divided by block of 30 minutes [10]
3. Construction of HTTP/HTTPS request/response pairs
4. Extraction of the features
5. Training of the classifier by using machine learning
6. Classification of traffic network

The detection method can be resumed as in Figure1 and the creation of the classifier as in Figure2.

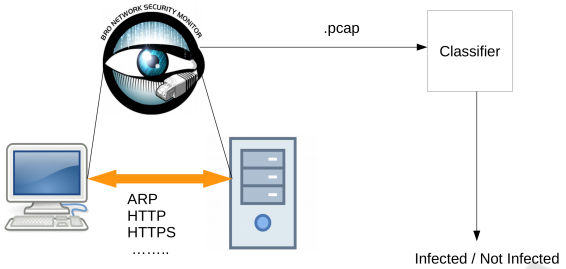

Figure 1: Malware Detection Protocol.

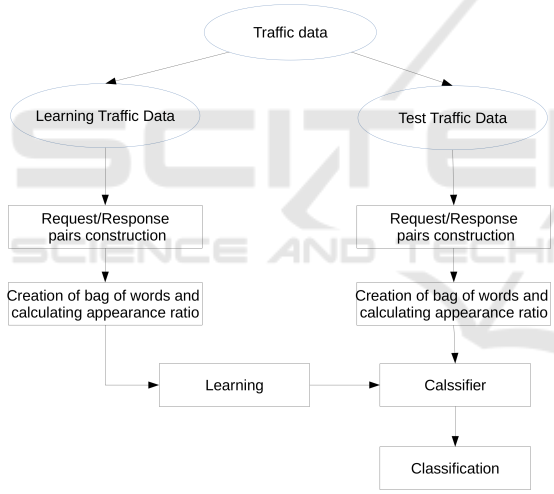

Figure 2: Classifier Construction Method.

4.2.1 Construction of HTTP/HTTPS Request/Response Pairs

The different network traffic captures are not directly analysed. In order to extract and construct the information Bro[11] was utilised. All the traffic captures are divided by request/response pairs in order to isolate communication. Request/response pairs correspond to all the traffic exchange between two machines (i.e packet with same source and destination address and the response to this packets). This is described in the Figure 3.

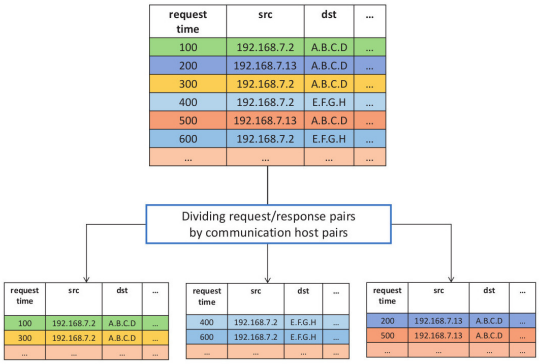

Figure 3: request/respons pairs(Ogawa et al.,).

It is really important to separate the data of every hosts pairs Indeed, if this is not done, we face the risk to have to much noise due to normal traffic data[12].

4.2.2 Creation of the Features

For HTTP/HTTPS packets different values are used as features for training our analyser. In this part will be explain how the data is created, and which features are chosen for both HTTP and HTTPS.
After having isolated the different host pairs, the fields of the packets are analysed. For every packets the content of the different fields, except for data field, are analysed. All of this data fields can take a finite number of values. The different values take by the fields during a period of time is the information that will be used. For every possible values of a field will be associated the frequency of appearance of this value. This frequency correspond to the number packets having this value divided by the total amount of packets (during the fixed period of time decided beforehand).
The fields that are analysed for HTTP and HTTPS are the following ones :

- **HTTP:** only the client headers are analysed, and the frequency of appearance of each client header is used[13].
- **HTTPS:** two elements characterised a HTTPS communication: a protocol and the encryption algorithm used by this protocol. We create a vector indicating the TLS version used during the exchange (TLS 1.0, 1.1 or 1.2), in this vector the en-

[10]This value of 30 minutes was choose ”randomly”. It could be interesting to determine the optimise time (or the optimise number of packets recived)

[11]The Bro Network Security Monitor https://www.bro.org

[12]In your infected computer there is not only malwares running but also your web browser, email browser...

[13]And it will be shown that it is sufficient to discriminate normal traffic to infected traffic.

cryption algorithm used is also indicated. The use of this two informations are used to discriminate the different HTTPS communications.

5 RESULTS

5.1 Values

In this section the values obtained for both HTTP and HTTPS will be given. The value used are the precision and recall for both normal and infected network traffic. The precision and recall obtain are relatively high. To obtain this results different machine learning were compared. We only give in this paper the best results obtained which was done by using AdaBoost on J48.

5.1.1 HTTP

The results obtained this time by only taking into account the content of HTTP client headers gave slightly better results than the previous study. It seems that the time interval and the other informations of the previous study add some noise that make decreases the precision of the detection[14].
The results show that most of the malwares can be detected by this method (more than 99%) however there is a small part of the normal network traffic (less than 7%) that is not correctly detected.
Our detection rate is similar to the most popular antivirus in the market[15] however our number of false positive is slightly higher[16] than most antivirus (vir,). However as 7% of normal traffic is considered as malware it can lead to an increase of threats, as it will decrease the reliance on the detection. Decreasing the number of false positive should be the next step to achieve in order to improve this method. Adding a white list for the false positives can also be an option.

Table 2: Average for HTTP using AdaBoost on J48.

	P_N	P_I	R_N	R_I
Average	0.94	0.99	0.93	0.99

5.1.2 HTTPS

The results obtained for detecting malwares using HTTPS are similar to the one for malwares using HTTP. The results were obtained by taking different information the TLS version alone, the cipher mode alone and TLS version and cipher mode together. The different models give, of course, different results but with similar values.
The model using the TLS version alone presents a very good value (more than 0.99) for the recall of infected traffic. However at the exception of true positive value they are all under 0.97 and with a false negative rate of 0.91 it cannot be used in real environment.
Using cipher mode only gives better results than the TLS version alone but still with a not so good value for the false positive rate (0.95).
By combining TLS version and cipher mode we obtain even better results than for virus using HTTP, surprisingly we obtain a better recall than with HTTP (0.97 against 0.93). The results are very close to the model using cipher mode alone. The false positive rate decreased and pass from 0.1 (for TLS vesion) to 0.03, which is more acceptable even if a white list is still necessary to avoid at maximum this situation.
TLS version and cipher mode give by themselves a lot of information concerning the nature of the network traffic analysed. This can suggest that virus do not behave as normal programmes even while communicating on Internet. It will be analysed on the following section.

Table 3: Average for HTTPS AdaBoost on J48.

	P_N	P_I	R_N	R_I
TLS version	0.97	0.97	0.91	0.99
Cipher mode	0.99	0.98	0.95	0.99
TLS version & cipher mode	0.99	0.99	0.97	0.99

5.2 Interpretation

The results detailed in the previous section show that we can obtain good results on detecting infected computer without having so much information on the sent packet [17]. That means the information given by every element is high. We analysed all this information to see how easy or not it is to discriminate infected machines from non infected ones. In order to obtain the information we look on the average of the average appearance of every elements for every host pairs. This statistics show that some values are a strong indicator of infection or not of a machine. This indicators will be analysed for both the HTTP and HTTPS study.

[14] that was already very high.

[15] 99.8% for the best one and 90% for the worst one.

[16] Except that it is difficult to compare as they give a number and not a percentage

[17] Two elements for HTTPS.

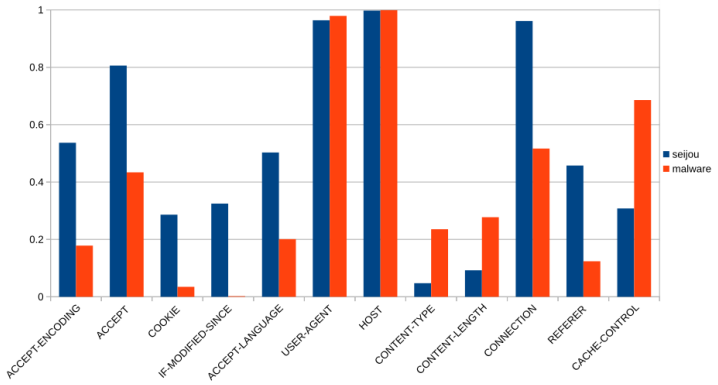

Figure 4: Average of client header for HTTP.

5.2.1 HTTP

As it can be observed on Figure4 representing the average appearance of client header for HTTP requests, normal and infected machines do not use so often every header.

The interesting headers are not the ones presenting the same values but header that are not used at the same way. Some of the headers that are not use as the same way by malwares will be listed and the interpretation given.

- **CONNECTION:** is a control option for the current connection. It should not be used for HTTP2. It is used more than 95% of the time by normal communication and more than 50% by malwares generated communication. The reason is that malwares used both HTTP1 and HTTP2 while most websites are still using HTTP1[18]
- **ACCEPT, ACCEPT-ENCODING, ACCEPT-LANGUAGE:** are used for defining the accepted contents and encoding for the response. It is used less than half of the time in malwares communication compare to normal ones. The accepted contents or encoding are generally known beforehand by the hacker. Making it not useful to use.
- **COOKIE:** contain the value of a cookie previously sent by the server. Used 30% of the time in normal traffic and less than 10% of the time by malwares. This is probably due to the reason that most of malwares only connect to a server and there is no need of recognising the user (that is the infected machine)[19]
- **CACHE-CONTROLS:** is used to control how the cache system works, and especially if the cache should be used for the packets received. With the data provided it seems that malwares tend to use more the possibilties of controling the cache with 70% of the packets received, than normal communications which reach only 35% of the packets.
- **IF-MODIFIED-SINCE:** a response is sent only if a modification happened. This is generally used to avoid loading content if no modification occurred, especially when there is a cache system. It is mostly not used by malwares.

5.2.2 HTTPS

Similar analysis than the one made on HTTP can be done on HTTPS. It will refer to Figure5 and Figure6.
First concerning the cipher mode used, for normal communication AES128 implementation using SHA-2 or SHA-3 are used. However for malware communication, a RC4 implementation using MD5 is used. This is very surprising when it is well known that MD5 is deprecated and no cryptographic security is given. Approximately 40% of the packets send in HTTPS by malwares are using cryptographic implementation using MD5. Blocking such packet could lead to suppress a big part of the malware traffic as it mostly not used in normal traffic. Using the exploit on such deprecated protocol could also be a possibility to discriminate malware traffic from normal one. This should be verified in another study.
Also, it happens that the cipher mode is not communicate or unknown[20][21], this is probably because hackers already know which algorithm will be used. But not giving this information gives a lot of information about the process sending this packets. As no normal traffic packet as been send with unknow field for the

[18] Less than 13% of the most visited websites use HTTP2. https://w3techs.com/technologies/details/ce-http2/all/all

[19] Or made by another way than sending cookie

[20] Or at list Bro IDS was not able to recognise the cipher mode

[21] not shown on the Figure

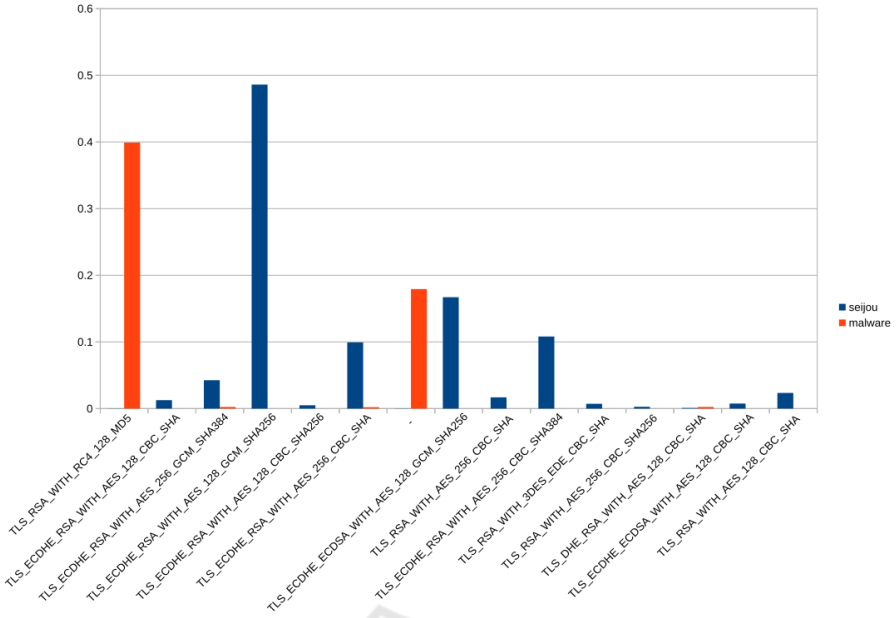

Figure 5: Average of cipher mode for HTTPS.

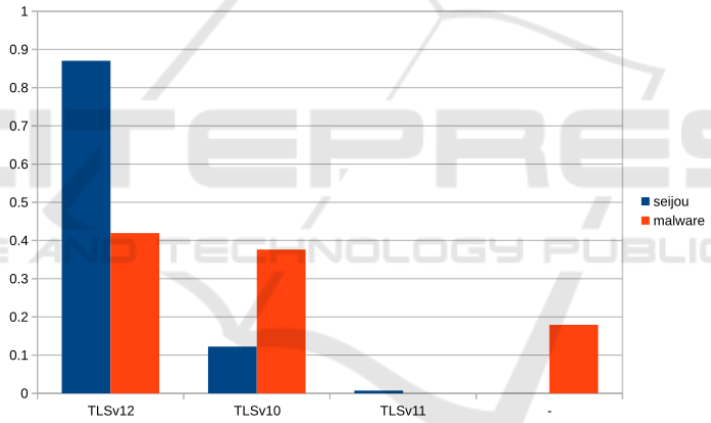

Figure 6: Average of TLS version for HTTPS.

cipher mode used, it gives a good way of detecting this kind of malwares.

A similar behaviour as been observed with the TLS version. Some of malwares do not indicate the TLS version used[22]. Giving the possibility to detect similar malwares. This detection can be done by indicating directly as suspect process sending packet and not precise the cipher mode or TLS version. The reason of this field not indicated should be study. Maybe this malware prentends to use TLS or use its own one .

5.2.3 Comparison with Cisco Results

As Cisco paper presents some techniques similar to ours, we will compare them. First of all Cisco paper present a very good detection of the normal files (near 100% and reach it under certain parameters but with high rate of false positive) which is far more better than our method as we have approximately 7% of false positive. For the detection of malwares the results is similar to our method but with different parameters. When they reach 100% for the normal traffic detection rate the malware traffic detection is low. However by decreasing this detection we obtain similar results for malware detection however they obtain

[22]And even sometimes modifying the value during the communication.

a better normal traffic detection[23]. By using two different methods we obtain similar results (with better precison in Cisco paper). This is probably due to the information added by TLS metadata.
One possibility of further work would be to extend our method by using TLS metadata in order to quantify the information given by it. It would be also interesting to test both method with the two datasets : ours and Cisco's one.

5.3 Malwares Do Not Respect Rules

The previous section show that malwares do not behave in the same way than normal process.
For HTTP traffics the header used are not the same and for headers shared by both type of traffic, the average frequency of using the headers is completely different. Normally headers are used to give information to web sites in order to make them adapt correctly the content for the client. But malwares know in advance, at least potentially, the characteristics of the C&C server they connect to. Malwares when they communicate with custom C&C server, do not communicate with website, hackers do not use this kind of headers probably because hackers already know, when programming malwares, which and how the information will be send. It is the same for HTTPS when TLS version or cipher mode are not precise. However for HTTPS it is even more surprising as malwares use mostly deprecated protocols or do not precise the encryption method and version as it is mandatory contrary to some HTTP headers. This is really easy to detect while observing network traffic packets. Because of that, an easy way to protect a system is to send and alert when deprecated protocol are used[24] and set a white list for old programs that cannot be changed and that use this kind of deprecated, and unsafe protocols.
However, this elements that can be used to detect malwares based on traffic analysis can be modified, especially for HTTPS. If hacker modified the way of constructing malwares by using classic cipher mode and TLS version for HTTPS and using more classic HTTP header, it will probably be necessary to use other features to detect malwares by this method. The following step would be to find features not based on the cipher mode used or easy modifiable header values and that describe more precisely the behaviour of malwares.

[23]But difficult to know exactly as we do not know what is the value, probably precision only, and the recall is not precised.

[24]As using deprecated protocol for cryptography it is really not recommended

6 CONCLUSION

Infected machine based on the analysis of the network traffic generated is relatively efficient. With more than 90% of precision and recall for both HTTP and HTTPS, with only few features. This is due to the fact that malwares are made to be only malwares and hackers already fixed how the data will be sent both by the malware and the C&C server. And also for some unknow reason they use deprecated protocol for SSL communication. However a modification of the packet sent could lead to the impossibility of detection by this method. There are different possibilities for future work. The first one is to find another features in the case of modification of the network behaviour by the hacker as explained before. The second is link to real environment implementation. The use for real time detection was not tested during this work. It would be of great use to see if it can be used in real time[25] and especially if it can detect malware that were not detected by classical antivirus. This method should be used complementary to other protection method. For further research, an analysis of the cipher suite used and the reason of why the data of TLS cipher suite and TLS version is not given during malware communication should be study.

REFERENCES

http://www.av-comparatives.org/wp-content/uploads/2014/04/avc_fdt_201403_en.pdf.

Detecting encrypted malware traffic (without decryption). https://blogs.cisco.com/security/detecting-encrypted-malware-traffic-without-decryption.

Ichino, M., Kawamoto, K., Iwano, T., Hatada, M., and Yoshiura, H. (2015). Evaluating header information features for malware infection detection. *Journal of Information Processing*, 23(5):603–612.

Maloof, M. A. *Machine Learning and Data Mining for Computer Security : Methods and Application*. London Spring.

Nasi, E. (2014). Bypass antivirus dynamic analysis. http://packetstorm.foofus.com/papers/virus/BypassAV Dynamics.pdf.

Ogawa, H., Yamaguchi, Y., Shimada, H., Takakura, H., Akiyama, M., and Yagi, T. Malware originated http traffic detection utilizing cluster appearance ratio. *ICOIN 2017*.

Otsuki, Y., Ichino, M., Kimura, S., Hatada, M., and Yoshiura, H. (2014). Evaluating payload features for malware infection detection. *Journal of Information Processing*, 22(2):376–387.

Page, C. R. E. (2003). Anti-debugging & software protection advice.

[25]and real environment not virtual machine