Detecting the Behavioral Relationships of Malware Connections

Sebastian Garcia
Czech Technical University, FEE
Karlovo namesti 13, 121 35 Praha 2
Prague, Czech Republic
sebastian.garcia@agents.fel.cvut.cz

Michal Pechoucek
Czech Technical University, FEE
Karlovo namesti 13, 121 35 Praha 2
Prague, Czech Republic
michal.pechoucek@agents.fel.cvut.cz

ABSTRACT

A normal computer infected with malware is difficult to detect. There have been several approaches in the last years which analyze the behavior of malware and obtain good results. The malware traffic may be detected, but it is very common to miss-detect normal traffic as malicious and generate false positives. This is specially the case when the methods are tested in real and large networks. The detection errors are generated due to the malware changing and rapidly adapting its domains and patterns to mimic normal connections. To better detect malware infections and separate them from normal traffic we propose to detect the behavior of the *group* of connections generated by the malware. It is known that malware usually generates various related connections simultaneously and therefore it shows a group pattern. Based on previous experiments, this paper suggests that the behavior of a group of connections can be modelled as a directed cyclic graph with special properties, such as its internal patterns, relationships, frequencies and sequences of connections. By training the group models on known traffic it may be possible to better distinguish between a malware connection and a normal connection.

CCS Concepts

•**Security and privacy** → *Malware and its mitigation; Intrusion detection systems; Security protocols;*

Keywords

Malware Detection, Behavioral Models, Groups of Connections, Machine Learning

1. INTRODUCTION

The detection of infected computers is usually problematic not because the malware patterns can not be detected, but because too much normal patterns are mis-detected [6]. This problem arises because the patterns in the network generated by malware are very similar to the patterns of normal traffic [7]. Moreover, it is known that normal connections usually use malware-like techniques, on purpose, to connect to their normal servers [5]. The problem is, then, how to better separate normal computers from infected computers given that the traffic of some connections looks very much the same.

Experiments with malware and normal captures have shown that most malware generates more than one connection [3], and that the connections seem to be coordinated in some way [1]. This suggested that a possible relationship between connections may be significant for a new detection method. The experiments done by the Stratosphere Project [2] have shown that it was already possible to model the behavior of individual connections for detection purposes. These detections were not exempt of errors, and those errors were the motivations for looking a detection alternative. The Stratosphere behavioral models are the base of our current analysis. We propose in this work to study the behavior of all the connections in a **group** simultaneously, in order to discover the properties of the group and to better distinguish the malware from the normal computers. We hypothesize that even though some malware connections can have the same behavior as normal connections, the group of connections itself must show intrinsic properties of the malware as a coordinated and logic unit.

There are previous analysis of groups in relation to malware. Most of these works grouped hosts to find all the infected bots, but they do not group connections from the same host. Moreover, they do not try to analyze the behavior of the *group*. The most related analysis has been the study of DNS requests. An example was the detection of malicious domain groups by using the temporal correlation of queries, given some well-known seed malicious domains [7]. The main difference with our work are (1) that they need a known malicious domain, and (2) that we analyze the behavior of the group. Another example analyzed the relationships between flows to form a graph structure of each host [4]. The authors summarized the behavior of a host in a vector in order to later cluster the malware families.

2. BEHAVIOR OF A CONNECTION

The base models for this work are the models created by Stratosphere. These models study which are the patterns of an individual connection in the network when all its flows are group together [1]. The stratosphere method only uses the flows in the network to create its models, and does not use the content of the packets. The base unit of work in what is called a *connection*. It is defined as all the flows that share

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

PrAISe '16, August 29 - 30, 2016, The Hague, Netherlands

© 2016 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ISBN 978-1-4503-4304-6/16/08...$15.00

DOI: http://dx.doi.org/10.1145/2970030.2970038

	Size Small			Size Medium			Size Large		
	Dur. Short	Dur. Med.	Dur. Long	Dur. Short	Dur. Med.	Dur. Long	Dur. Short	Dur. Med.	Dur. Long
Strong Periodicity	a	b	c	d	e	f	g	h	i
Weak Periodicity	A	B	C	D	E	F	G	H	I
Weak Non-Periodicity	r	s	t	u	v	w	x	y	z
Strong Non-Periodicity	R	S	T	U	V	W	X	Y	Z
No Data	1	2	3	4	5	6	7	8	9

Symbols for time difference:

Between 0 and 5 seconds:	.
Between 5 and 60 seconds:	,
Between 60 secs and 5 mins:	+
Between 5 mins and 1 hour:	*
Timeout of 1 hour	0

Figure 1: Criteria to assign a letter to each flow in a connection in the Stratosphere project. The letter represents a state.

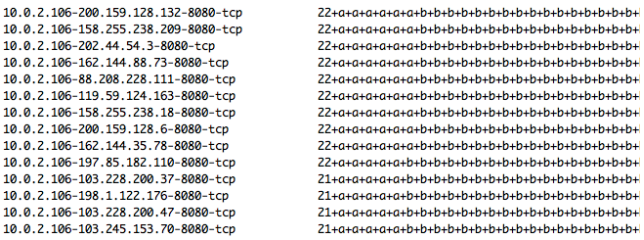

Figure 2: Geodo malware group behavior. It can be seen how Geodo generated several identical connections to its C&C servers. Moreover, the connections where generated in a very specific order that revealed its group behavior.

the following key: source IP address, destination IP address, destination port and protocol. All the flows matching the same key are grouped together. A connection focus on the behavior of a user against a specific service.

Stratosphere assigns a behavioral pattern to each connection. To assign it, each flow is seen as a new *state* of the connection. Each flow, then, is is assigned a letter to represent its properties. To assign a letter to each flow (to each state), three features of each flow are used: its size, its duration and its periodicity. Based on these features we can select a letter to each flow. The purpose of using letters is that a security analyst could be able to better recognize patterns and therefore verify the detections. The strategy to assign letters to flows is shown in Figure 1.

Each flow that arrives is analyzed to see to which connection it belongs. Then the three features are extracted and a letter is assigned. The connection, then, has a string of letters that identifies its own behavioral pattern. These behavioral patterns are the ones used in this work for the analysis of the behavior of groups.

3. PROPERTIES OF A GROUP OF CONNECTIONS

In the previous Section it was shown how Stratosphere models the behavior of a unique network connection. The experiments with Stratosphere found that malware usually has several connections that can be groped together. These connections exhibit a strong relationship between them, which allows an expert to detect patterns. To show these relationships, two different malware and one normal connection were analyzed. The malware are a Geodo binary with MD5 7cd95f1500289d4ba938169610060de0 [1], and an Upatre binary with MD5 b8fc436bab8c22cdfa2d4d137358ca93 [2]. The Normal connection belongs to a unique computer in a University network [3]

The group behavior of the first malware, Geodo, is shown in Figure 2. It shows how it generated fourteen connections to different C&C servers, each of them having the same behavioral pattern (*22+a+a+a+a+*). The group behavior of Geodo can be seen when the patterns are visualized in time, like in this accelerated video that can be downloaded [4]. The video shows that one flow is sent to each connection in turn following a round-robin schedule. Therefore, there seems to be a process generating the flows in this pattern according to a certain logic. Apart from the original pattern of each connection, there is now a pattern for the group.

The second example corresponds to the Upatre malware shown in Figure 3. It generated 46 connections to different C&C servers. The connections have a similar pattern than Geodo, sending a flow to the next connection in a round-robin schedule. The most important difference with Geodo is that the Upatre connections have three distinct behavioral patterns: Some connections show the pattern *S*S,*, some show the pattern *s*s*s*b*b*b** and other the pattern *h*h*h*h*h**. The group behavior can be seen better in an accelerated video [5]. This video shows the round-robin scheduling of the flows and the potential relationship between them. There seems to be some logic in the generation of the flows, which we know came from the same binary malware.

The behavior of the normal capture is easier to interpret because there are no clear indications of groups. Figure 4 shows the connections of the normal capture. It can be seen that the computer is active and sending traffic, some of it as a result of normal work, but there are not clear groups of connections to create. Moreover there are no periodic connections. It is possible, however, that a groups of related connections could be found with a different criteria. If we analyze this capture in an accelerated video [6] it can also be seen that there are no special patterns to recognize.

4. BEHAVIOR OF THE GROUP

Based on the patterns shown in the previous Section we can analyze now how to represent the behavior of the group. To create the group behavior it is first necessary to identify which connections belong to the group. The cases of Geodo and Upatre were selected because there is a clear relationship: the connections came from the same malware binary, so they are based on a programming logic. In some sense, the idea is to model the logic of the malware by observing its network connections. In Geodo and Upatre the group is easy to identify because they have very similar individual

[1]Download from https://mcfp.felk.cvut.cz/publicDatasets/CTU-Malware-Capture-Botnet-122-1/

[2]Download from https://mcfp.felk.cvut.cz/publicDatasets/CTU-Malware-Capture-Botnet-162-1/

[3]Download from https://mcfp.felk.cvut.cz/publicDatasets/CTU-Normal-6-filtered/

[4]https://youtu.be/zBJsdm03UDA

[5]https://youtu.be/Z8-100svkQs

[6]https://youtu.be/owYxtgV2y7I

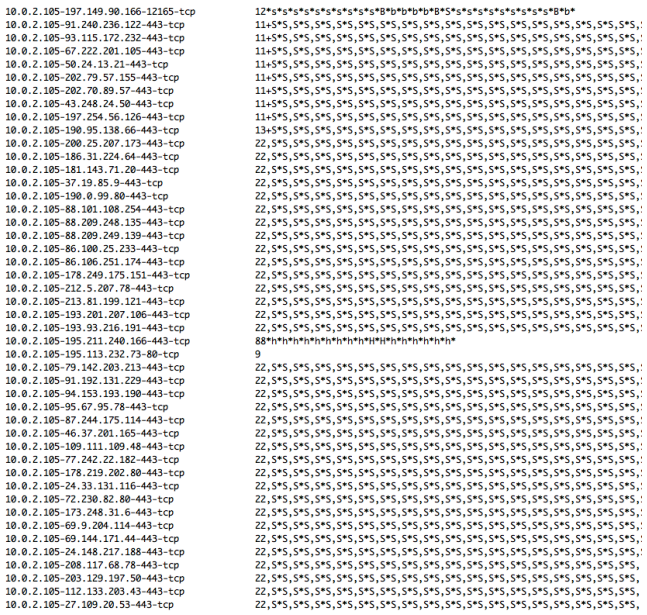

Figure 3: Upatre malware group behavior. Upatre generated several identical connections to its C&C servers. The connections where generated in a specific order that revealed its behavior.

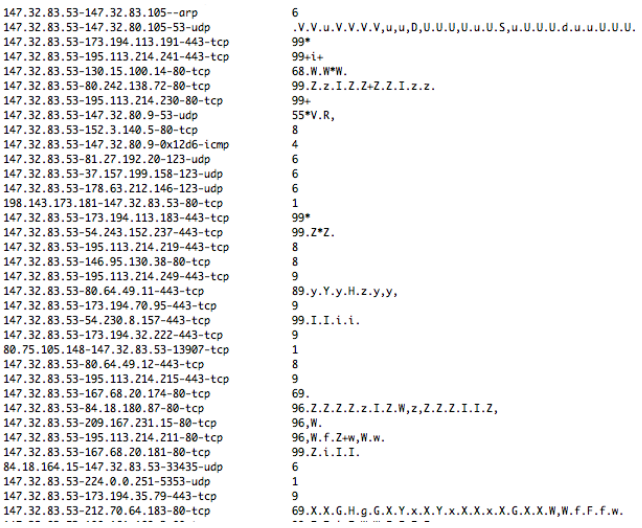

Figure 4: Behavior of a normal computer in a University network. The connections have different patterns and none is periodic. It is not easy to group connections since there are no clear relationships. This image only shows the significant connections, see the video for a complete recollection.

patterns, they use the same destination port, and they were made almost simultaneously. We could identify two ways of creating the groups: (1) group the connections that have the same individual behavioral pattern of letters, (2) group the connections that have a relationship based on the destination IP address (including the WHOIS information). Both of these techniques would group the malware connections and would also group some normal connections. We use the first way of creating the groups based on the similarity of the individual behavioral patterns.

To analyze the behavior of the group it is necessary to analyze its components. Each group is composed of the connections on it, identified as C_i, and the flows arriving to all the connections, identified as F_j. From each flow it is possible to extract: its arrival time, its duration, its size and the connection to which flow belongs to. Each time a flow arrives there is a *transition* from the last connection that received a flow to the next, and therefore a sequence: C_1 -> C_2 -> C_3 -> C_1. Based on C_i, F_j and this last sequence, the following components of a group are computed:

- A directed graph, identified as D computed from the previous sequence.
- The amount of connections.
- The Stratosphere behavioral pattern of each individual connection computed from F_j.

The use of a directed graph gives us a structure to analyze the group behavior. More information is needed to complement it, such as the frequencies of the transitions in the graph. Each directed graph D has its own properties:

1. If D is a cyclic graph, i.e. if it is a loop.
2. The amount of cycles performed by the flows.
3. The frequency of the loop.
4. The frequency of sending a flow to the next connection in D.
5. The frequency of sending a next flow to the *same* connection in D.
6. The delay between the last and first connection in the group.

An example graph of the Geodo group behavior is shown in Figure 5, with its fourteen connections. This graph shows how each new arriving flow goes from one connection to the next connection in the group.

The properties of D can then be analyzed based on trained malware and normal captures to find the behaviors that best separate them. In particular the first restriction for being considered a group behavior is that the graph should be cyclic. If there is no repetitive pattern, then there is no group behavior. Another restriction is that there should be a minimum amount of cycles to be detected as malware, since it indicates that the loops are not by chance. The frequency of the loop and the frequency of sending a flow to the next connection may be a characteristic of each botnet and therefore may be trained. The frequency of sending a flow to the same connection is already incorporated into the Stratosphere behavior of that connection. Finally, the delay

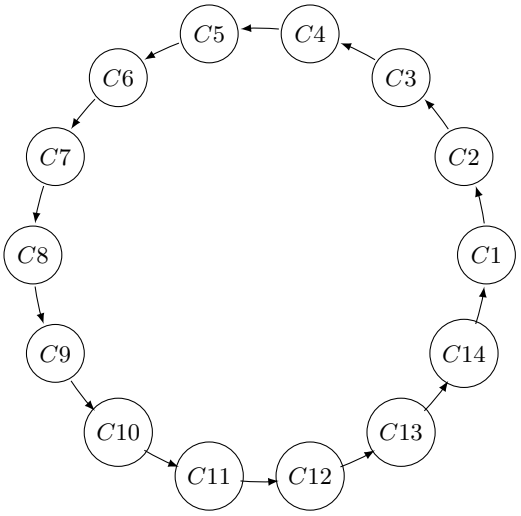

Figure 5: Directed Cycle Graph for the Geodo malware group behavior. There are 18 connections, and the sequence is a round-robin schedule.

to restart the loop is also characteristic and can be changed during the lifetime of the malware.

Based on these properties it is possible to analyze the malware. It should be considered that the Geodo malware showed two distinctive patterns, *b+b+b+* and *b*b*b*b**, and therefore some of the values changed for the group behavior. The properties of Geodo group are:

- Amount of connections: 14
- Amount of cycles performed: 848
- If the graph D is cyclic: Yes.
- Frequency of loop: 3 minutes, 20 seconds (*b+b+b+* pattern).
- Frequency of loop: 17 minutes 20 seconds (*b*b*b*b** pattern).
- Frequency of sending a flow to the next connection: 10 seconds without considering the restart of the loop.
- Delay between the last and first connection: 1 minute 10 seconds (*b+b+b+* pattern).
- Delay between the last and first connection: 15 minutes 10 seconds (*b*b*b** pattern).
- Frequency of sending a flow to the same connection: 3 minutes, 20 seconds (*b+b+b+* pattern).
- Frequency of sending a flow to the same connection: 17 minutes, 20 seconds (*b*b*b** pattern).

The example of the Upatre malware is more complicated, given that the C&C connection *10.0.2.105-197.149.90.166-12165-tcp* is used in two groups with different connections but the same properties. The properties of both groups are:

- Amount of connections: 24
- Amount of cycles performed: 72
- If the graph D is cyclic: Yes.
- Frequency of loop: 33 minutes.
- Frequency of sending a flow to the next connection: 21 seconds
- Delay between the last and first connection: 17 minutes.
- Frequency of sending a flow to the same connection: 33 minutes.

These values for the properties of the malware groups show how different they are and how characteristic they can be. The case of the Normal capture was easy in this example because it didn't have any group of connections that could be distinguished by our method. This means that if we use the detection method of the group behavior it may be possible to differentiate the malware infections from the normal traffic. We acknowledge that more normal connections are needed for a better testing. The identification of group properties in these examples suggest that they may be a good model of behavior for a detection method.

5. CONCLUSIONS

This position paper proposes that by studying the behavior of a group of connections it may be possible to improve the detection of infected computers. The motivation was that some normal connections look like malware connections and they are not easy to distinguish. We proposed that connections can be grouped together according to how they are related, and those groups can be analyzed to find their properties, sequences and behavioral patterns. The properties of a group seem to have enough differential power to be able to separate normal and botnet computers. Malware may change to avoid any technique, but the behavioral changes may be too costly for them. This research will continue by formally defining the properties of groups, experimenting with larger datasets and using the groups to detect infected computers.

6. REFERENCES

[1] S. Garcia. Modelling the Network Behaviour of Malware To Block Malicious Patterns . the Stratosphere Project : a Behavioural Ips. In *Virus Bulletin*, number September, pages 1–8, 2015.

[2] S. Garcia, M. Grill, J. Stiborek, and A. Zunino. An Empirical Comparison of Botnet Detection Methods. *Computers & Security*, 45(0):100 – 123, 2014.

[3] S. Garcia, V. Uhlir, and M. Rehak. Identifying and Modeling Botnet C&C Behaviors. In ACM, editor, *Proceedings of the 1st International Workshop on Agents and CyberSecurity (ACySE '14)*, New York, 2014. ACM.

[4] S. Nari and A. A. Ghorbani. Automated Malware Classification based on Network Behavior. pages 642–647, 2013.

[5] M. Patterson. Security Vendors Teaching Bad Actors How to Get Past Firewalls, 2015.

[6] R. Sommer and V. Paxson. Outside the Closed World: On Using Machine Learning for Network Intrusion Detection. *2010 IEEE Symposium on Security and Privacy*, 0(May):305–316, 2010.

[7] H. R. Zeidanloo and A. B. A. Manaf. Botnet Detection by Monitoring Similar Communication Patterns. *International Journal of Computer Science and Information Security*, 7(3):36–45, 2010.