Behavioral Study of Bot Obedience using Causal Relationship Analysis

Pekka Pietikäinen
Oulu University Secure Programming Group
University of Oulu
FIN-90014 Oulu, Finland
pp@ee.oulu.fi

Lari Huttunen
NBI Finland IT Crime Unit
PO Box 285
FIN-01301 Vantaa, Finland
first-conference@itcu.fi

Abstract

Botnet discovery can be difficult, since the existence of a network is often discovered only after it used for widespread activity such as a DDoS or a phishing scam. Sharing intelligence on a potential botnet traffic is also problematic mainly due to data privacy issues.

In this paper, we describe some currently used methods for identifying botnets and issues which arise when applying them in practice. We will identify the types of information that could be shared between different stakeholders and the technical means available to gather such data. Finally, we will present causality graphs and describe initial experiences in applying them to analyzing botnet incidents.

Contents

1 Introduction

Botnets are collections of software robots, or bots, centrally controlled using e.g. the IRC protocol. They have become significant, but hidden, part of the Internet. Botnets vary greatly in size, with small networks being only tens of strategically placed hosts and the largest ones being hundreds of thousands of hosts. [1]

The life-cycle of a typical botnet victim consists of the following phases:

- Infection either directly over the network or indirectly through user interaction
- Trojan payload is downloaded and/or executed
- Bot joins the botnet, typically using IRC as the Command & Control (C&C) channel
- Bots are used for some activity (DDoS, further exploitation etc.)
- Bots are updated with a new version or new business functionality through the operator issued update command

Detecting botnets proactively is very difficult. Typically, the networks are only discovered after they are used for widespread activity, such as a DDoS attack, where the sheer quantity of traffic makes the existence of the network clear cut. There are both active and passive methods for discovering active botnets [2].

This paper is structured as follows. In Section 2, we describe different methods that are currently used to discover botnets. In Section 3, we describe the basic concepts behind causal relationship graphs and how they apply to botnets. In Section 4 we present some results of preliminary analysis. Finally, conclusions are drawn in Section 5.

2 Detection Methodologies

There are a variety of data sources that can be used for detecting botnet activity. There are two basic types of detection mechanisms, active and passive. Active detection of malware is done by attempting to catch live instances of running botnets trying to propagate either directly by exploiting a vulnerable network service or indirectly through user interaction, i.e. through malicious spam email or web page.

Another way of analyzing botnets is through passive means, where suspicious traffic is captured from the network. Botnets have some common characteristics, which may be used to find them. Other ways of classifying botnet detection mechanisms are through their

- *Scope:* How widely can they be used to track botnet infections, from individual hosts to multi-organization networks,
- *Detection time:* When does the method identify new botnets, early (before the botnet has propagated) or late (botnet has already caused a massive infection),
- *User:* What kind of user is the method is used by, and
- *Type:* Is the method Direct (Method measures botnet activity directly), or Indirect (Side-effects of botnet activity are measured).

Currently used methods for detecting botnets are summarized in Table 1. We will now look at the different methods in more detail.

Data source	*Scope*	*Detection time*	*User*	*Type*
Victim	Individual machine	After infection	Unhappy end-user	Direct, Indirect
Honeypot or spampot	Varies	Early	Security researcher	Direct
Antivirus software	Individual machine	Infection attempt	End-user, network operator	Direct
IDS with signature	Network	Infection attempt	Network operator	Direct
IDS without signature	Network	After infection	Network operator	Indirect
DNS-based IDS	Network	After infection	Network operator	Indirect
Flow data	Several networks	Early to postmortem	Network operator	Direct, Indirect

Table 1: Different botnet detection methods

2.1 Data Sources

As stated above, the range of data sources for detecting an active botnet vary in scope, latency and interactivity. By this we mean the traditional dichotomy seems to be that of host-based detection and network-based detection. Both of these methods are not necessarily clear-cut, since host-based mechanism may trigger network based alerts and network based mechanisms may be distributed over several networks.

2.1.1 Victims

Unfortunately the first detector of a botnet infection is often the end-user, whose computer has been compromised and turned into a zombie. Discovering the control mechanism from the compromised computer is a good lead to investigate a given botnet but often the box will yield little more than that in terms of finding out about the criminal activity it has been used for.

2.1.2 Honeypots and Spampots

Honeypots and spampots are a widely used approach for collecting new types of malware. Traditionally, an unpatched honeypot is placed on the network and closely monitored for infections [3]. Honeypots have the benefit of gaining detailed information on the operation of the piece of malware, including obtaining the Trojan payload and monitoring the botnet C&C traffic.

New methodologies such as modular low interaction honeypots are emerging. A good example of such a tool is Nepenthes [4], which emulates well known vulnerabilities and interacts with the worm trying to spread to the honeypot. Spampots collect malware by harvesting email spam to find new, previously unknown instances. These approaches, however, will most likely only discover the low hanging fruit, i.e. the malware with aggressive spreading behavior or known exploit mechanisms.

Honeypots, however, have drawbacks. First of all, it requires the honeypot to become infected. When the infection occurs through a worm which attempts to attack as many hosts on the Internet as possible, special attention must be paid to egress filtering of the

traffic generated by the infected host. Second of all, operating a honeypot may pose legal problems in terms of data privacy and liability issues. Third of all, bots may contain antiforensic capabilities, such as honeypot detection mechanisms. Nevertheless, properly operated honeypots will provide valuable intelligence on bot software and the C&C channels they use. The skillset and effort required to maintain an effective honeypot and disassemble the malware limit their usefulness to security researchers.

2.1.3 Anti-Virus Software

Anti-Virus software aims to stop malware by matching the signature of malicious activity as changes to the operating system and/or its network connectivity. Once the signature is matched, the normal procedure seems to be to attempt to quarantine the malicious code and to notify the computer owner or a central AV management console. In a corporate environment, this can yield in a heads-up for the systems administrators, but unfortunately AV engines can detect only malicious code which has been identified as such. Evaluating the usefulness of AV software as an information source for botnet investigations depends largely on the particular deployment.

2.1.4 Intrusion Detection Systems

Passive methods collect data from the network and do not interact with the possible exploitation mechanism communicating over the medium at hand. Most of the methods to detect attacks are signature-based, which means that they detect only malicious activity which the heuristic has been tuned to detect. In terms of botnets, most of them still use IRC as the control mechanism; however, other protocols such as P2P protocols are in the process of being adopted.

Given the fact that IRC still predominates, many attempts have been made to incorporate the control commands as triggers for botnet control traffic. These signatures, however, are very easy to go around, since customizing the command language into something, which the signature will not match, is not very difficult. In addition, IRC controlled bots have legitimate uses as well, which yield false positives – not to mention the legitimate IRC traffic on the network generated by human communication.

Also, detecting botnet traffic from network captures is becoming more and more difficult, since botnets are using ephemeral port numbers for their IRC servers and some botnets are encrypting the C&C traffic. This kind of evolution will require the passive detection mechanism to be able to identify secondary features of bot infection such as propagation or attack behavior detection [2].

2.1.5 DNS-based IDS

A new promising type of IDS for botnets uses analysis of DNS queries to find misbehaving hosts [5]. This relies on the fact that botnets typically use DNS to find the IP address of the controller, which allows the controllers to quickly be moved to new hosts as previous ones are disconnected.

A DNS-based IDS looks for anomalous DNS queries and logs them. The anomalies can be i.e. known botnet controllers, abnormally popular queries or clients, or queries

with non-regular qtypes, such as large numbers of MX queries for a server that does not run a SMTP server.

A problem with DNS-based IDS's is that they are very susceptible for false-positives and thus are not conclusive enough for marking bad hosts with a 100% certainty. Therefore additional information, such as NetFlow data should be used to find compromised hosts and controllers.

Another approach is to implement a passive DNS replication infrastructure to log queries and their responses without causing overt privacy issues, since only the queries and their responses are logged, not the individual hosts generating the queries. This technique was introduced by Florian Weimer at FIRST 2005 [6]. The benefits of the DNS logging infrastructure, however, will be reactive unless a detection mechanism is incorporated into this approach. The balance between privacy and security once again does not necessarily go hand in hand.

2.1.6 Flow data

Often the only source of information available to an organization about a botnet infection is NetFlow data gathered on the traffic crossing the network border. NetFlow contains summary data for each flow of traffic traversing a network router. There are several different formats used to encapsulate NetFlow data. The most recent version of NetFlow [7] is an extensible format, which currently defines 89 field types (e.g. MPLS labels, IPv6 addresses and AS numbers associated with the data). Older versions are more limited in the information they can provide, however for behavioral analysis this is usually enough. Table 2 shows what kinds of fields typically are available in a NetFlow record.

Start time	End time	Source interface	Source address	Source port	Destination interface
Destination address	Destination port	Protocol	TCP Flags	Packets	Bytes

Table 2: Types of data available in a typical NetFlow record

This information could be used to identify the existence of a botnet within a given organization. Effective analysis, however, will require correlation of flows between organizations, which often is not possible due to technical and legal reasons. Even storing the data, let alone sharing it, may be problematic, since for large organizations it means additional backups of gigabytes of data on a daily basis. In addition, the sheer volume of traffic at large sites is often so great that gathering flows is only possible through sampling.

Isolating the botnet traffic from regular traffic (and possibly anonymizing it) makes sharing the data possible from a legal point of view. A potential possibility for address anonymization is the Cryptography-based Prefix-preserving Anonymization algorithm (Crypto-PAn) [8]. It provides a method for anonymizing IP addresses in a way that has several benefits. It is prefix-preserving, that is unanonymized addresses with a k-bit prefix will share a k-bit prefix when anonymized. The anonymization is also cryptography-based and consistent across traces (if same cryptographic key is used). CANINE [9] is a tool for applying this algorithm to NetFlow data.

Address anonymization causes additional difficulty when analyzing the data, as the same C&C hosts will likely appear in several traces. Therefore, a mechanism for querying whether two anonymized addresses are the same (without revealing the actual identity) is required.

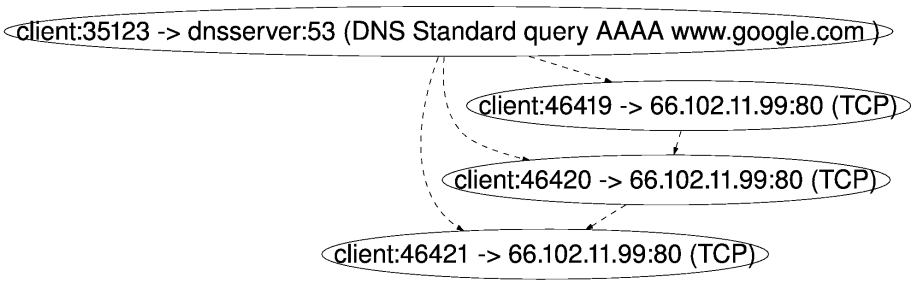

Figure 1: WWW query represented as causality graph

When botnet incidents are investigated, isolating the relevant traffic can be a major effort. In this paper, we introduce a framework which utilizes causal relationships in NetFlow data and flow anonymization techniques which should allow effective dissemination of botnet intelligence between different organizations.

3 Causality Graphs in Network Traffic

Causality graphs are a method for modeling and visualizing interactions between the components of a multi-network distributed system. The basic principle behind building causality graphs is representing communication between two processes as an event, instead of the traditional process-centric model, where the events would be one process sending data and another receiving it. These are interrelated causally, i.e. hosts a and b communicating with each other followed by b and c communicating would result in a causality graph of $(a \leftrightarrow b) \rightarrow (b \leftrightarrow c)$. An example of a causality graph is shown in Figure 1, where a client retrieves a web page (http://www.google.com/). This results in a DNS query and three TCP connections. Previously we have used packet captures from multiple networks as the data source [10], however the technique applies to flow-level data as well.

Causality graphs provide a powerful tool for grouping potentially related events together. The data can then be analyzed using graph-theoretical methods, such as finding isolated subgraphs to find traffic that is not interrelated.

As the quantity of traffic grows, so does the causality graph. Communication patterns are a technique for abstracting and separating known traffic patterns from causality graphs. For instance, a simple communication pattern might be a TCP connection from a to b leading to an ICMP Administratively Prohibited error from an intermediate router c.

4 Experimental results

To validate how causality graphs can be used to analyze botnet incidents, we analyzed anonymized flow data from a real-life botnet infection of a Finnish organization. The data was stored in ASCII and anonymized using CANINE by the organization providing us with the data. Anonymization was done on the IP addresses only, the timestamps and port numbers were unchanged. The ASCII file format was chosen to ensure that no accidental information leak would occur (some binary NetFlow formats support including partial packet payloads). Sharing this data without anonymization would have been totally

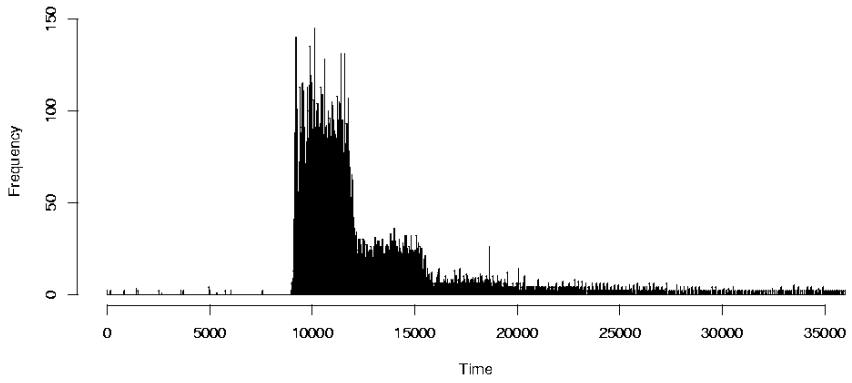

Figure 2: Activity on controller port

impossible for legal reasons. The data was collected from a border router, thus traffic internal to the organization is not seen in the data.

The goal of the analysis was to isolate traffic related to the botnet from regular traffic, identify compromised hosts and controllers, and find what behavioural similarities the victims shared.

The raw flow data was originally 7.5 gigabytes, which corresponds to 10 hours of traffic and 62.4 million flows. This was imported into a SQLite database to allow structured searches and minimization of interesting subsets of data to be subjected to causality analysis and visualization. In practice it turned out to be practical to work with subsets of data, since processing and visualization data sets of this size are very resource intensive and not interesting from an investigative point of view. Table 3 summarizes the results of the analysis.

Total distinct addresses:	8293953
Total flows:	62393760
Control port distinct addresses:	650
Control port flows:	18269
C&C hosts	6
C&C flows:	18157
Number of victims:	546
Victim flows:	23753270
Control port flows:	17892
Port 445 flows:	23484991
Other traffic:	250387

Table 3: Summary of traffic within the dataset

The number of addresses represented in the flow data is very large, over 8 million hosts, thus some narrowing down of the data to potential victims and controllers is required. The easiest way of doing this is identifying the C&C port. As Figure 2 highlights, there is a radical increase in C&C port traffic roughly 9000 seconds after the beginning of the dataset, which corresponds to the beginning of the infection at the organization in question. The analysis could also start with an individual victim, which in this case is very easily detectable from a very large number of port 445 probes being sent to the outside world.

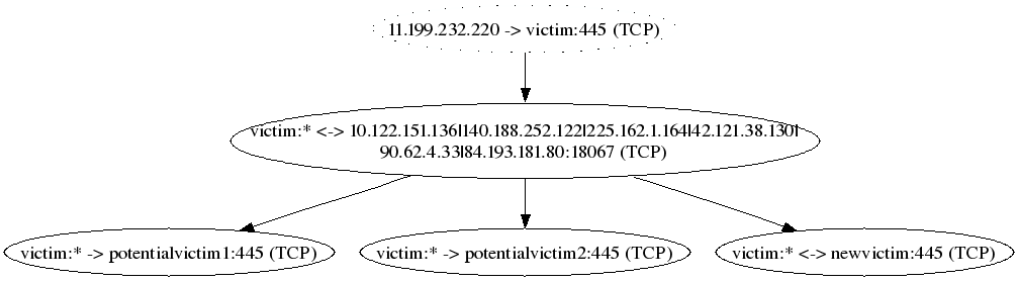

Figure 3: Causality graph for botnet victim

Using traffic C&C traffic as the baseline reduces the amount of flows to be processed significantly. There were only 650 hosts that used the C&C port for something. Some of these were merely by coincidence (the botnet used an ephemeral port number, and thus some flows using the port were e.g. legitimate HTTP connections with the C&C port randomly assigned as the source address). This is where the concept of causality really helps. Victims of the same bot infection share similar behavioral patterns. The victims connect to a small set of C&C hosts, and as commands are issued by the bot herder through the C&C channel, the zombies execute the commands, which in turn results in new netflows (updating to new software versions, DDoS etc.).

In our example case, the zombies quickly started propagating using a MSRPC vulnerability. In fact, over 38% of the total flows were propagation attempts. This allowed easy identification of the victims. Going back in time and finding what hosts used the C&C port allowed identification of the C&C servers. There were 6 C&C servers and 546 victims (and 98 hosts which were not involved in the botnet despite using the same port number). The initial infection would be a causal predecessor to the C&C connection, however in this case the infection was from the inside, and thus would not show up in NetFlow data. Interestingly enough, in just minutes before the initial infection, an outside host (anonymized IP 11.199.232.220) performed a MSRPC sweep over the network, but this was blocked by a firewall (resulting in one-way flows to the victims in their causality graphs). A simplified causality graph for a botnet victim is shown in Figure 3 (in reality the victims attempted to infect thousands of hosts on the internet). Only some infection attempts succeeded, which are seen as bidirectional port 445 flows where several kilobytes of data were transmitted.

Isolating the control traffic was easiest to accomplish by identifying the C&C port, which in turn called for lead type information at the outset of the investigation. In practice, combining DNS logging with flow data analysis is more likely to cut down the data mass substansively rather than working with plain causality analysis. It is also much easier to start an investigation by discovering a single host or suspect DNS record rather than trying to separate ”slashdotters” from botnet control traffic as a whole for example. Once this is achieved, however, causality analysis on a subset of flows will easily identify:

- victim hosts in a given network (or many networks if this information is shared),
- botnet controllers, which normally exhibit diversity in terms of geographic location, and
- dropsites, which provide the update mechanism or business functionality.

5 Summary and Future Work

In this paper, we have described some of the currently used methods for analyzing botnets. There probably is no silver bullet for dealing with botnets. In addition to the most important mitigation technique, fully patched systems, the currently available analysis methods complement each other. Ideally, data from several of the available techniques should be correlated with each other to deal with the threat, starting from individual hosts and ending with backbone routers.

Correlating data between organizations poses privacy concerns, which must be addressed. Address anonymization algorithms alleviate this problem, but can make effective analysis more difficult. Combining several methods to yield lead type information on possible malicious activity can in turn help law enforcement to focus their attention on the more disruptive botnets, which isn't necessarily measured in quantity of traffic rather than monetary damage inflicted by the criminal business run on the botnet platform. A number of bot capabilities for criminal activities have been detailed in the paper published by CERT/CC in 2005. [1].

A possible procedure for handling botnets could work as follows:

- Passive DNS replication infrastructure is deployed across organizations to yield leads on possible botnets.
- Each organization participating in information sharing analyzes their flow data regularly to find botnet-like traffic
- Flows, which constitute possible botnet traffic are anonymized and transferred to a third party
- Data from different organizations is correlated. If the existence of a botnet is confirmed, the participants are notified.

When the flow data is prepared to be used as evidence, the quantity of flows unrelated to the incident can be minimized. This is done by only including flows with a distance of n from the C&C traffic. The evidentiary value of flow data analysis especially across organizations will in turn help proving the case in court, since the number of victims can be enumerated and the monetary value of direct damages put forth by the victims can be more easily justified. All in all, NetFlow analysis is often an overlooked resource for investigating this type online crime commited en masse over the Internet. If only more organizations gathered flows at their network perimeter.

References

[1] N. Ianelli, A. Hackworth, "Botnets as a Vehicle for Online Crime", CERT Coordination Center, December, 2005. Available: `http://www.cert.org/archive/pdf/Botnets.pdf` (Accessed: May 10th 2006)

[2] E. Cooke, F. Jahanian, D. McPherson "The Zombie Roundup: Understanding, Detecting, and Disrupting Botnets". Steps to Reducing Unwanted Traffic on the Internet Workshop (SRUTI '05), Cambridge, Massachusetts, USA.

[3] The Honeynet Project & Research Alliance "Know your Enemy: Tracking Botnets: Using honeynets to learn more about Bots", 2005. Available: http://www.honeynet.org/papers/bots/

[4] Nepenthes - versatile tool to collect malware. URL: http://nepenthes.sourceforge.net/ (Accessed: December 10th 2005)

[5] A. Schonewille, D-J van Helmond, "The Domain Name Service as an IDS: How DNS can be used for detecting and monitoring badware in a network", Research report, University of Amsterdam, February 5, 2006, Available: http://staff.science.uva.nl/~delaat/snb-2005-2006/p12/report.pdf (Accessed: May 10th 2006)

[6] F. Weimer, "Passive DNS Replication", April 2005, Available: http://www.enyo.de/fw/software/dnslogger/first2005-paper.pdf.

[7] Cisco Systems, "Cisco IOS NetFlow Version 9 Flow-record Format" Available: http://www.cisco.com/application/pdf/en/us/guest/tech/tk362/c1550/ccmigration_09186a00800a3db9.pdf (Accessed: May 10th 2006)

[8] J. Fan, J. Xu, M. H. Ammar, S. B. Moon (2004) "Prefix-Preserving IP Address Anonymization", Computer Networks, Volume 46, Issue 2 , 7 October 2004, Pages 253-272, Elsevier.

[9] Y. Li, A. Slagell, K. Luo, and W. Yurcik, "CANINE: A Combined Converter and Anonymizer Tool for Processing NetFlows for Security" International Conference on Telecommunication Systems - Modeling and Analysis (ICTSM) , Dallas, Texas, November 17-20, 2005.

[10] P. Pietikäinen, J. Röning (2005) "Communication Pattern Extraction: Visualizing causal relationships in complex systems", 20th International Conference on Computers and Their Applications, New Orleans, LA, March 16-18, 2005.