An Integrated Framework for Malware Collection and Analysis for Botnet Tracking

Rakesh Kumar Sehgal
C-DAC,Mohali

D. S. Bhilare
Head, IT Centre, Indore

Saurabh Chamotra
C-DAC,Mohali

ABSTRACT

The paper presents the design of an integrated malware collection and analysis framework for botnet tracking. In proposed framework we have used Honypots as malware capturing tool. The proposed system design is unique in the sense that the information regarding the configuration of honeypot on which malware sample has been captured is saved with malware sample in the malware data-base. This system configuration information saved with the malware sample is used at the time of dynamic malware analysis for creating malware execution environment. As an execution environment thus created is analogous to environment in which malware was captured therefore it generates true expected execution behavior leading to capturing of accurate execution traces. Further we have demonstrated the effectiveness of the proposed solution with the help of a prototype system.

Keywords

Culture, Productivity, Social Networks, Workplace, Malware, Hack

1. INTRODUCTION

Cyber attack trend has undergone a great shift from attacks targeting to disable big I.T infrastructures to those that target common people. Hence now a day cyber attacks are not just restricted to big IT infrastructures but also compromise computers located at homes, schools and government offices around the world [3]. The pool of such compromised machines are further used by the attackers to perform activities such as spamming, DDOS, information stealing etc [4].Such pools of compromised machines under the control of a single attacker are known as botnets.

The compromised machines in these botnets are called ‘Bot’[24]. As the compromised machines in a botnet are analogous to robots which are slaves controlled by a master, hence the term bot is used to refer these compromised machines. Similar to a robots these bot machines are controlled by a master system which is known as botmaster or command and control server [20][9][16].

These command and control servers or the C&C server use internet as a medium to communicate with the bots. Using internet these C&C servers sends commands to their bots and receives information from them. One can estimate the impact of such botnet based attacks by the fact that a normal botnet may have number of bots computers ranging from few thousands to millions [23].

Besides that all of the bots of a botnet may not be limited to a particular region or network and rather they are actually scattered throughout the internet. Hence in order to track threat posed by such geographically distributed infected machines one needs to collect and correlate data from multiple data sources [12]. In the approach followed by us we are using Distributed Honeynet system as a tool to collect, detect and track these botnets. Integrated malware collection and analysis framework proposed by us is based upon the Honeynet technologies. Honeypots are the information system resources whose value lies in being attacked and probed [33].

Honeynet is a network of such Information system resources. These Honeynets play a critical role in defending against the new cyber attacks and threats by producing first hand and focused information regarding these attacks[1][2].

The integrated collection and analysis framework design proposed by us is based upon the idea of involving the contextual knowledge in the process of malware analysis. The work presented in this paper is based upon following premise:

- For the detection and tracking of botnets one needs to collect and correlate data from multiple data sources located in different network environments.
- The contextual information regarding the environment in which malware has been collected should be incorporated in the malware execution to enable dynamic malware analysis process.

In the work presented in this paper we propose a Honeynet based collection and analysis framework for tracking IRC and HTTP botnets [10]. We have demonstrated the effectiveness of the proposed solution using a Distributed Honeynet prototype system. The results have shown that the proposed system is able to track real-world botnets with high accuracy.

2. RELATED WORK

Worldwide efforts have been made for the detection and early warning of possible cyber threats using various distributed monitoring systems. Being geographically distributed, these distributed monitoring systems widen the scope of monitoring and data collection. Various projects like mwcollect[30],lurree.com[29],Hive[17],Network telescope[26], Noha[31] and Honeynet consortium[32] have used the concept of distributed monitoring[25].

These projects have adopted approaches like monitoring the unused IP space [26], large scale deployment of low interaction Honeypots [29], clusters of Honeypots known as Honeyform [27] or the networks of Honeypots that are Honeynets [11] for monitoring purposes. Even the data sets these projects collect form the monitoring resource varies; as some collects network traffic some firewall and IDS logs[28] and even the malware samples[17][30].

One common thing in all these projects is; they all collects malicious data and perform analysis on it for doing threat prediction.

Among all these projects, lurre.com [7] has first time introduced the concept of enrichment of captured malicious data with contextual information regarding the observed data. They called this data meta-data [29] and use this data at the time of analysis. This concept is similar to ours idea of enrichment of the malware sample with the contextual information regarding the environment in which sample was captured. We intend to use this information at the time of analysis to render better analysis results. The difference between our work and the work of luree.com is in the dataset. They use malicious traffic as dataset where as we are focused on the malware samples.

In area of malware collection good work has been done in [20] where a toolkit is developed by Jianwei Zhuge et.al which they have named as Honeybow.

This toolkit provides an automated solution for the capturing and collection of the autonomously spreading malwares. Although the toolkit is completely based upon high interaction Honeypots, the main strength of this toolkit is its capturing mechanism which incorporates varied range of capturing techniques.

Another project HIVE [17] has taken a leap ahead and had developed an open source software based framework for collection and analysis of malware samples. The uniqueness of framework proposed by HIVE is its design which incorporates a combination of high interaction and low interaction honeypots. Although the approach proposed by HIVE is better in terms of system downtime and honeypot diversity but still the degree of dynamism offered in the honeypot configurations is limited [13][14].

In the work presented in this paper we have proposed an integrated system for malware collection and analysis which is very much similar to that of HIVE. The uniqueness that we have introduced is the incorporation of system configuration information on which the malware sample was been captured in the analysis process.

At the time of malware collection from honeypot we capture the system configuration information in terms of operating system, services running and software loaded on that Honeypot. We fuse this data with malware binary sample and convert it in to a relational database format. This information hence collected is used at the time of the dynamic malware analysis for the creation of execution environment.

3. MALWARE COLLECTION FRAMEWORK

To define the scope of malware collection we have categorized malwares in two categories [5][6].

- The one which requires external medium to propagate i.e. email worm, driven by download malwares etc.
- Others which autonomously spread by first scanning the internet for available target and then compromising them by exploiting the system vulnerability.I.e code red, nimda, blaster, morris etc[6][18].

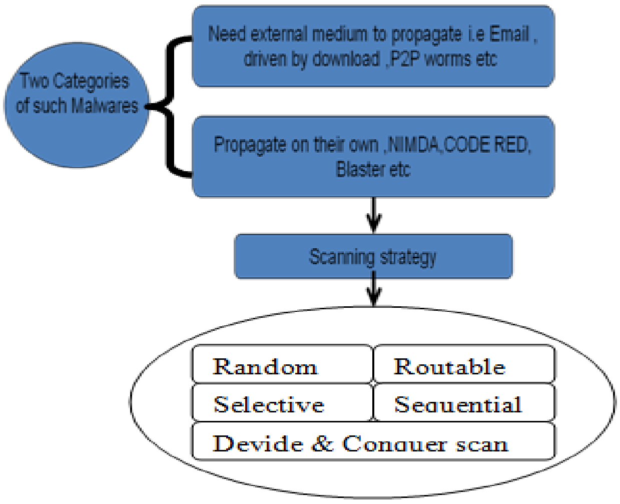
Two Categories of such Malwares
Need external medium to propagate i.e Email , driven by download ,P2P worms etc
Propagate on their own ,NIMDA,CODE RED, Blaster etc
Scanning strategy
Random
Routable
Selective
Sequential
Devide & Conquer scan

Figure 1: Malware Types

The collection framework developed by us addresses to the collection of second class of the malware samples those spread autonomously without the help of any external medium. As now viruses that infect using external medium pen drive, email worms, or driven by download type of malware attacks are not considered in current scope hence this characterization has helped in focusing the scope of collection. Malware class that is focused in current scope, first scans the internet using various scanning strategies, discovers online vulnerable machines, then exploits the vulnerability in those machines [19][5].The Distributed Honeynet system developed by us for capturing the autonomously spreading malware is structured on three tier architecture. As shown in the figure 2

Layer1 incorporates Honeynet sensors which captures the malware samples and sends collected data to the central Distributed Honeynet server on a regular basis.

Layer2 incorporates Distributed Honeynet server which performs activities such as registering new nodes, processing data sent by remote nodes, data fusion and converting the data in to a relational data base format.

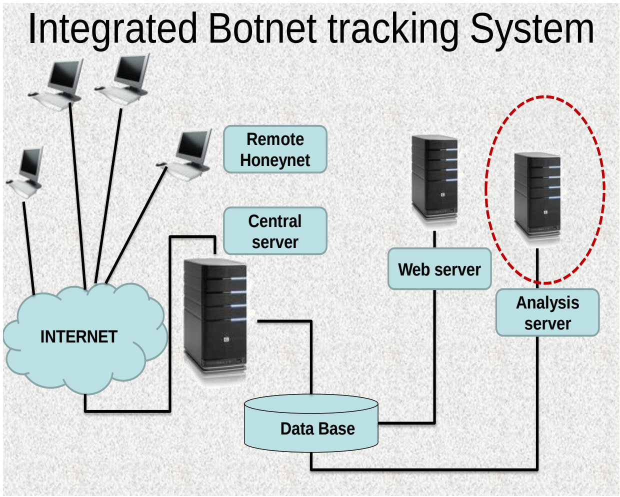

Figure 2: Three tier architecture

Layer 3 consist the Database which acts as a data source for analysis engine.

The figure 3 shows the network diagram of Distributed Honeynet implementation. The prototype system has four nodes deployed in different ISPs. Each node consists of a combination of a low interaction and high interaction Honeypot.

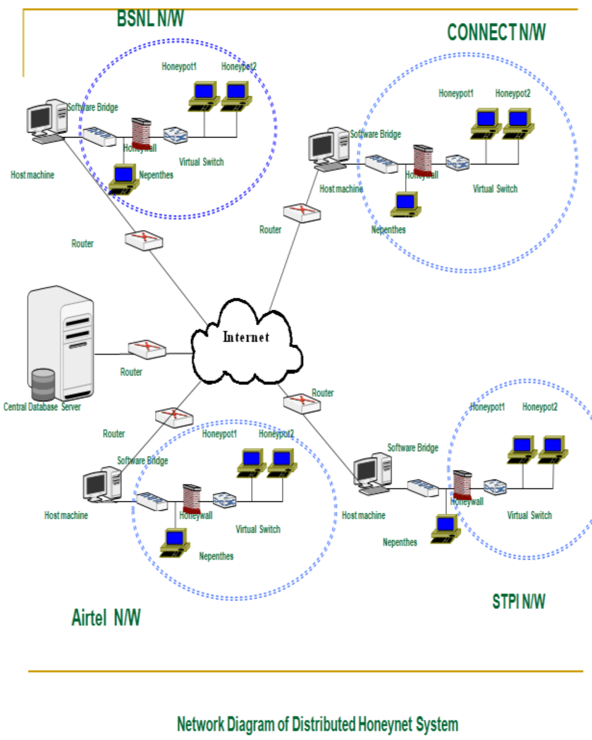

Figure 3 Network Diagram

4. MALWARE ANALYSIS FRAMEWORK

The malware binary collected from the Honeypots is fused with the data regarding the system configuration on which it is captured and then this fused data is converted in to the relational database format. Figure 4 shows an abstract picture of the malware database. The master table has malware sample, its MD5 hash and sourceHP as its attributes.

The sourceHP field is a foreign key in master table which acts as a primary key in the table 2. Table 2 performs the mapping between the Honeypot and the system configuration that was loaded on it.

The details of the system configuration in terms of operating system, running services and installed software are stored in table 3. Similarly the collected malware samples with the related system configurations are saved in the central database. During the analysis phase this database is accessed and the meta data saved is used in the dynamic malware analysis process.

We have integrated Honeysand[21] a sandbox environment developed by us for performing dynamic malware analysis. Honeysand is open source tools based sandbox environment which is specifically designed for bot detection and botnet tracking. Inputs to Honeysand module is the malware sample and the related configuration Information

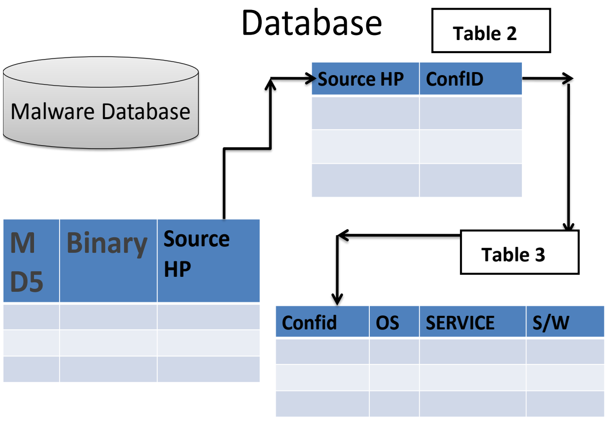

Figure 4 Malware Database

The figure 5 shows an over view of the integrated malware analysis system**.** The malware fetcher module fetches the malware sample with the meta.This information is given as an input to the Honeysand system. Honeysand uses the system configuration information to generate an execution environment. Once the execution environment is ready Honeysand executes the malware sample in the execution environment and collects the network logs and the Native API call sequences. The API call sequences are used by the bot detection engine. The Bot detection engine is a module which uses the API call sequence mining technique used in [22] for the identification of the bot binaries. If a malware binary sample is labeled as the bot binary by the bot detection engine the botnet tracking engine searches its network traffic for botnet related command and activities. Using this information Botnet tracking engine creates a network fingerprints for each bot binary sample. Such a fingerprint contains following attributes:

- DNS information
- C&C commands (IRC OR HTTP)
- egg download source IP

Using this network fingerprint the botnet tracking engine performs the clustering of the bot samples. Every unique cluster hence created represents a unique Botnet.

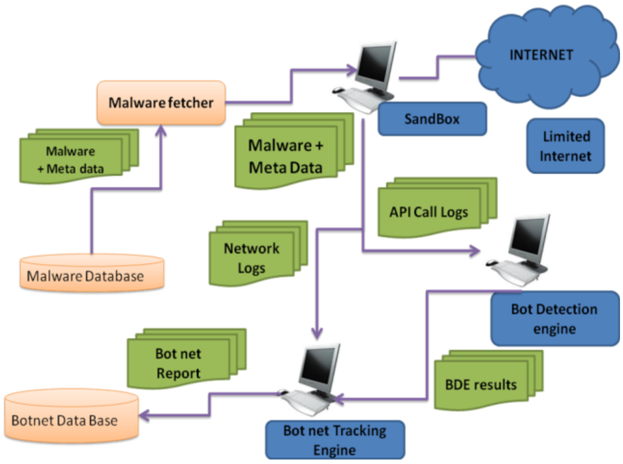

Figure 5 Malware analysis framework

5. EXPERIMENTAL RESULTS

The table 1 shows the count of the malware samples collected from the different nodes of the Distributed Honeynet system. These malwares are further processed as their MD5 hashes [8] are calculated. Based upon the MD5 hash it was observed that out of the malware samples shown in table 1 the malware sample shown in table 2 were the unique malwares samples collected.

Table 1: Malware sample collected

Distributed Honeynet Nodes	Malware count
Node 1	3709
Node 2	741
Node 3	2310
Node 4	1502

Further after performing dynamic malware analysis on the collected malware samples the classification results obtained are shown in table 3.

Table 2: Unique Malware samples

Distributed Honeynet Nodes	Malware count
Node 1	264
Node 2	133
Node 3	155
Node 4	151

Based upon our classification we have classified the malware samples in to Bot and Not Bot classes. The Not Bot class contains bot, worms, backdoors, Trojans and undetected malware samples.

Table 3: Malware sample classification

Node	Bot	Not Bot
Node 1	95	169
Node 2	31	93
Node 3	54	101
Node 4	71	80

We have also performed an experiment to show the effect of incorporating the contextual information on the results of dynamic malware analysis. In this experiment we have first executed the 1200 binary samples in an sandbox environment without considering the system configuration on which they were been captured. Out of these 1200 malware samples only 255 were successfully been executed and shown some system traces. In other case we executed the 91 malware samples in the Honeysand environment considering the system configuration on which they were been captured. Out of those 91 malware sample 90 malware samples were been executed successfully and the one that wasn't executed was due to the virtualization detection. The results of the experiment are presented in table 4.

Table 4 Malware execution results

Sr. No	Binary Submitted	Actual Binary Executed	IRC C&C server	HTTP infected source
1	1200	255	22	60
2	91	90	29	12

Table 5 shows the results of the clustering performed by the botnet tracking engine the IP addresses of the C&C servers are sanitized

Table 5 C&C server IP detected

C&C SERVER	BOTNAME	Type
60.x.x.100	B354,B375,B356,B523,B56,B550,B361,B522,B263,B586,B316,B283,B394,B495,B534 B575	IRC
208.x.x.101	B103,B104,B106,B108,B109,B139,B147,B148,B197,B201,B141,B143,B145,B202,B151,B152,B14,B19,B17, B1	HTTP
208.53.xx.101	B100,B111,B112,B184,B115,B14,B151,B152,B141, B143,B145,B1,B201,B202,B179,B17	IRC
124.124.xx.42	B543,b361,b33,b113,b138, b8	IRC
208.53.xx.101	B427,b498,b298,b550,b337,b557,b353,b551,b497,b121,b547	HTTP

Further table 6 shows the commands that were used by the corresponding C&C servers.

Table 6 Bot commands

C&C	Token Found
60.x.x.100	**PING,NICK,JOIN,USER,MODE**
208.x.x.101	**HTTP/GET**
208.53.xx.101	**PING,NICK,JOIN,USER,MODE**
124.124.xx.42	**PING,NICK,JOIN,USER,MODE**
204.53.xx.10	**HTTP/GET**

6. CONCLUSION

In the work presented in this paper we have presented the design of an integrated framework for the collection and analysis of the malware samples for botnet tracking. In proposed framework we have introduced the idea of saving the system configuration on which the malware was captured with the malware sample in the database for using it in analysis process. This information is used at the time of dynamic malware analysis for the creation of an ideal execution environment. Further for giving a proof of concept we have created a prototype system of this integrated framework with four distributed nodes deployed in four different ISPs and integrated Honeysand an open source tools based sandbox environment as an analysis engine in this framework. The result generated using the above prototype system is presented in the paper and it has shown that considerable improvement in the detection and tracking of the botnets could be achieved using the proposed framework.

7. REFERENCES

[1] John Levine, Richard LaBella, Henry Owen, Didier Contis, Brian Culver "The Use of Honeynets to Detect Exploited Systems Across Large Enterprise Networks" School of Electrical and Com puter Engineering

[2] Vinod Yegneswara,Paul Barford,Vern Paxson"Using Honeynets for Internet Situational Awareness"

[3] http://securityresponse.symantec.com/avcenter/

[4] http://www.caida.org/analysis/security/witty/

[5] Cliff Changchun Zou, Lixin Gao, Weibo Gong, Don Towsley "Monitoring and Early Warning for Internet Worms"University of Massachusetts at Amherst

[6] David Moore, Vern Paxson, Colleen Shannon, Stuart Staniford, Nicholas Weaver "The Spread of the Sapphire/Slammer Worm",2003

[7] Leurre.com: on the Advantages of Deploying a Large Scale Distributed Honeynet Platform

[8] www.viruslist.com/de/viruses/encyclopedia?chapter=152540403

[9] A. W. Jackson, D. Lapsley, C. Jones, M. Zatko, C. Golubitsky, and W. T.Strayer, "SLINGbot: A system for live investigation of next generation botnets," in Cybersecurity Application and Technologies Conference for Homeland Security (CATCH), Washington, DC, USA, Mar. 2009.

[10] J. Goebel and T. Holz. Rishi: Identify bot contaminated hosts by irc nickname evaluation. In USENIX Workshop on Hot Topics in Understanding Botnets (HotBots'07), 2007.

[11] Reto Baumann and Christian Plattner, "White Paper: Honeynets", 26 February 2002

[12] J. Yang, P. Ning, X. S. Wang, and S. Jajodia. Cards: A distributed system for detecting coordinated attacks. In SEC, 2000

[13] Iyad Kuwatly, Malek Sraj, Zaid Al Masri, and Hassan Artail. "A Dynamic Honeypot Design for Intrusion Detection" American U. of Beirut

[14] Christopher Hecker, Kara L, Nance, and Brian Hay" Dynamic Honeypot Construction "

[15] X. Jiang and D. Xu. Profiling self-propagating worms via behavioral footprinting. In Proceedings of CCS WORM , 2006

[16] F. Freiling, T. Holz, and G. Wicherski. Botnet tracking: Exploring a root-cause methodology to prevent denial-ofservice attaks. In ESORICS'05.g"

[17] Davide Cavalca and Emanuele Goldoni HIVE:an Open Infrastructure for Malware Collection and Analysis

[18] J. Zhuge, T. Holz, X. Han, C. Song, and W.

[19] Zou. Collecting autonomous spreading malware using high-interaction honeypots. In ICICS 2007, pages 438–451, 2007. [19] M. Garetto, W. Gong, D. Towsley, "ModelingMalware Spreading Dynamics," in Proc. of INFOCOM 2003, San Francisco, April, 2003.

[20] Liu, P. W. and Tyan, H. R, "An Adaptive defence mechanism for P2P Botnet." Unpublished doctoral dissertation, Department of Information and Computer

[21] Saurabh Chamotra, Mr.Rakesh Kumar Sehgal, Dr. Raj Kamal "Honeysand: An Open Source Tools Based Sandbox Environment for Bot Analysis and Botnet tracking"

[22] Hengli Zhao, Ning Zheng, Jian Li, Jingjing Yao, Qiang Hou" Unknown Malware Detection Based on the Full Virtualization and SVM" 2009 International Conference on Management of e-Commerce and e-Government

[23] P. Barford and V. Yegneswaran. An inside look at botnets.In Proc. Special Workshop on Malware Detection, Advancesin Information Security, 2006

[24] Trend Micro. Taxonomy of botnet threats (white paper),November 2006

[25] Saurabh Chamotra, Rakesh Kumar Sehgal Dr. Raj Kamal ,J.S.Bhatia" Data Diversity of a Distributed Honeynet based malware collection system" ,Emerging Trends in Networks and Computer Communications (ETNCC), 2011 International Conference

[26] D. Moore. Network telescopes: Observing small or distant security events. In 11th USENIX Security Symposium, Invited talk, San Francisco, CA, Aug. 5–9 2002. Unpublished

[27] L. Spitzner. "Honeypot Farms", Infocus, Aug. 2003. http://www.securityfocus.com/infocus/1720.

[28] DShield. Distributed Intrusion Detection System, www.dshield.org, 2007

[29] C. Leita , V.H. Pham , O. Thonnard , E. Ramirez-Silva ,F. Pouget , E. Kirda , M. Dacier , The Leurre.com Project: Collecting Internet Threats Information using a Worldwide Distributed Honeynet 2008 IEEE DOI 10.1109/WISTDE.2008 WOMBAT Workshop on Information Security Threats Data Collection and Sharing

[30] Mwcollect http://alliance.mwcollect.org.

[31] Details of NOHA project: http://www.fp6-noah.org/publications/presentations/moeller-tfcsirt17.pdf

[32] Honeynet Project http://www.honeynet.org/

[33] L. Spitzner, Honeypots- Tracking Hackers, Indianapolis, IN: Addison-Wesley, 2003, pp. 242-261