A Framework for Botnet Infection Determination through Multiple Mechanisms Applied on Honeynet Data

Sanjeev Kumar, Rakesh Kumar Sehgal, Saurabh Chamotra
Cyber Security Technology Division
Centre for Development of Advanced Computing
Mohali, India
Email:{sanjeev|rks|saurabhc}@cdac.in

***Abstract*—Botnets are a class of internet attacks having different characteristics as compared to the normal internet attacks. One of the features that uniquely characterize a botnet attack is that "the infected machine (Bot) is being remotely controlled by an entity called "Botmaster". The Botmaster remotely controls these infected systems through "Command and Control" servers (C&C).**

Over a period of time complexity of botnets has increased many folds. Advance techniques employed by botnets such as protocol encryption, complex botnet structure and multi stage infection propagation models have made the botnets detection a challenging problem. Hence there is a need of a botnet detection mechanism which is independent of C&C protocol, structure and the infection propagation model used by the botnet.

In the work presented in this paper, the experiments have been performed for evaluating the strength of existing botnet detection techniques, and proposed an advance detection mechanism which uses a logical combination of existing open source solutions with Honeynet technologies and our own mechanisms for botnets detection. A test setup as a proof-of-concept of the proposed framework with the experimental results is presented in this paper.

Keywords— Malwares, Botnets, Network Security, Intrusion Detection System, Honeynet.

I. INTRODUCTION

Botnets have become a great threat to the internet users as most of the cyber-attacks such as Distributed Denial of Service (DDoS), flooding spams, phishing mails as well as stealing user's passwords, credit card numbers etc. are launched using botnets. As the size of a botnet increases the severity of the attacks launched by the botnet also increases. Hence the prime objective of a botnet is to grow exponentially by using the exploited system to further exploit vulnerable systems and recruit them as attacking agents. These efforts of botnet to grow have resulted in exponential raise in Botnet based attacks, attracting the attention of research community. Many Researchers are working to develop mechanisms for detection and mitigation of the botnet attacks

A Botnet itself is a network of compromised machines controlled by Botmaster. The botmaster uses control and command servers which registers, synchronizes and communicates with the Bot systems. **Figure 1** shows various protocols employed by the command and control server for Bot communication.

These botnets further uses different network topologies in combination with above mentioned network protocol. Some of popular network topologies employed by the contemporary botnets are centralized, distributed hybrid and randomized.

Figure 2 depict the complete anatomy of a botnet and visualize the following three key aspects of botnets:

1) Infection propagation Mechanism
2) Botnet Malicious Behavior
3) Communication protocol

Each of the above mentioned aspect of the botnet vary in different botnet families and hence characterizes a botnet family.

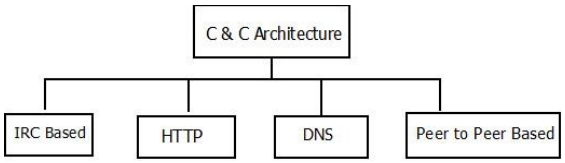

Fig. 1. Command and Control Architecture

For survival, a botnet has to look for new victims, exploit them and recruit them in its bot army on a regular basis. This botnet infection cycle is termed as infection propagation model of the Botnet. The Infection propagation mechanism is a key feature characterizing the botnet families. As different Botnet families employs different infection propagation models (i.e. server side exploits, malicious emails, Infected USBs, client side code exploitation through malicious web contents etc.).

978-1-5090-0210-8/16 $31.00 © 2016 IEEE
DOI 10.1109/CICT.2016.12

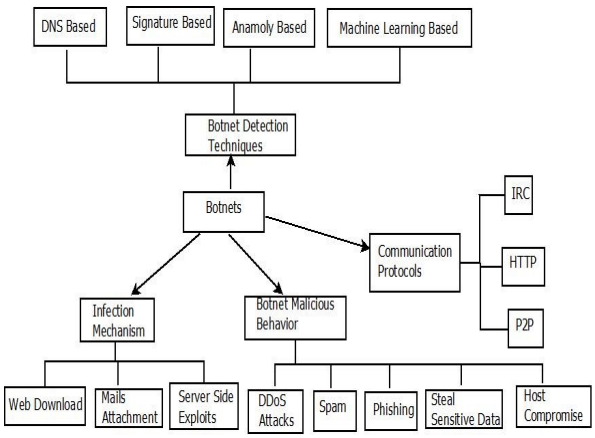

Fig. 2. Botnet Terminology

Post successful infection communication with command and control server using IRC/ HTTP/ P2P protocol is established. Command and control server on the other hand registers the new Bot systems and timely sends the commands for enhancements, synchronization and to perform malicious activities (i.e. phishing, spamming, key logging information stealing etc.) to the registered Bot systems.

In the work presented in this paper, the experiments have been performed for the evaluation of the detection strength of existing popular detection mechanism. We have tested these mechanisms in the Honeynet environment and based upon the experimental results we have proposed a framework for botnet detection. Proposed framework is a logical combination of the existing available measures with detection mechanism developed by us.

II. RELATED WORK

Botnet detection is an interesting problem which has been addressed by many researchers [22-24]. They have identified some of the basic traits (i.e. remote control behavior [18] [21], infection life cycle [19] need to coordinate and communicate [20] etc.) for botnet characterization and used them as feature sets for the development of Botnet detection mechanism. During the literature survey we came across the following two type of feature sets used for the detection of botnets.

1) Network level: Which uses features such as network flow data, IDS alert sequences, DNS queries etc.
2) System level: Whish uses features such as API call sequences, malware analysis etc.

A number of open source tools has been proposed and developed using features from either of the above two classes (i.e. Bothunter [19], Botswat [18], Rishi [20], and Botminner [27] which is an improved version of Botsniffer [26]). As it is always an arm race between the attackers and the defenders hence attackers also have devised approaches for evading the available detection mechanisms. These approaches uses techniques such as protocol encryption, complex network structures, stealth coordination and communication, Fast Flux [25], server migration and dynamically generated domain names [30].

We tested these detection tools with the Honeynet data and it was observed that no single tool can successfully detect all the botnet families. In other words as Botnets are very diverse in their modus of operandi, structures and the targets hence making it difficult to develop a single common approach for the detection and mitigation of all the existing botnet classes. In the work presented in this paper we put forward this hypothesis that no single detection approach can detect all classes of Botnets and the existing detection mechanisms can logically combined to complement their detection capabilities.

III. FRAMEWORK DESIGN

In this section, the design of botnet detection framework is presented. Data feeds for the system is obtained from Honeynet system. Honeynets acts as a victim machine luring attackers. Once compromised, it starts capturing activity logs of the attack. In case if attack is a Botnet attack, Honeynet also captures communication dialogues with the remote C&C server.

In established honeynet infrastructures, different OS and service combinations are configured. Network traffic in the form of PCAP dumps and malware dropped on honeypots are captured. The network traffic dumps are processed a *posteriori* through the existing publically available open source botnet detection tools to get the botnet attributed such as command and control server, infection source, protocol used etc. whereas the Malware samples are first processed through the popular Antivirus engines. If labeled as Bots then a developed script maps the labeled bot malware with corresponding network PCAP dump. The extracted PCAP logs are submitted to the analysis engine and details of botnet command and control server, infection source, no. of bytes exchanged, volume of data transferred corresponding to the bot sample are extracted.

A. Approach for Measurement of Botnet Traffic

There are various approaches available for the botnet detection such as malware code analysis, behavioral analysis using sandbox execution [28] [29] [22]. In case of Honeynets every packet entering and leaving is logged with corresponding system level activities on a regular basis. Hence Honeynet effectively captures and logs all the phases of bot life cycle (i.e. registration, updating the system by downloading new malwares, launching attacks for infection propagation) once got infected by a botnet infection. The length and the breadth to which the Honeynet is able to capture the botnet activity couldn't be achieved through sandbox execution. Hence Honeynets are the ideal mechanism for analysis of the Botnets. We have exploited this fact to develop Honeynet based framework for botnet detection. The system uses post

Honeynet exploitation data as an input for the botnet detection. [1]

B. Honeynet Implementation Scenario

The deployment scenario for honeynet deployment is directly facing the internet having public IP address. The significance of direct public IP is making the honeypots sensors more exposed to the internet. Further to make the Honeypots deployed in Honeynet visible to the attacker's public IP addresses are assigned to them. To make the Honeynet deployment hassle free, easily manageable and easily recoverabl, the virtualization technology [2] [3] for Honeypot creation is used. Figure 3 illustrates attack data collection framework which is a heterogeneous mixture of low interaction (i.e Dionaea) [6] and high interaction Honeypots (i.e. WinXPSP2, WinXPSP3, Window7 etc.).

Data logging: The Honeypots logs the data at network level and the system level. At the system level binary files dropped on honeypots are captured through file system integrity checker and at network level the network traffic is logged through the TCPDUMP [7] tool.

Data control: The data control mechanism controls the traffic to and from honeypot machines. We have used an improved version of the conventional data control mechanism Honeywall [8] which was used in the GEN III Honeynet architecture. The data control mechanism used by us has two level of security.

The first level of security is provided by the IPTABLES [9] firewall and the second level of security is provided by Snort-inline [10] reverse firewall with subscribed signatures set. The data control mechanism allows all the inbound connection towards the honeypot from outside world but selectively allows packets form the honeypot to the outside world.

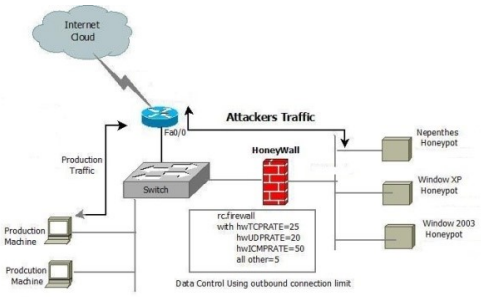

Fig. 3. Honeypots facing the Internet

IV. PROPOSED ANALYSYS FRAMEWORK

If we look at the history of botnets, IRC protocol was the first and most commonly used botnet communication protocol. Table 1 lists various botnet families and the communication protocol used by them. As per literature Agobot, SDBot uses IRC as a communication medium [4, 5] whereas botnet families such as Zeus, Spy Eye and zero access employs HTTP protocol. The reason behind popularity of HTTP protocol is that in most of the secured networks connection form inside to outside world using any protocol except HTTP protocol is not allowed. P2P botnets are distributed in nature and there is no concept of central command and control server, every host in botnet communication acts as C&C server. Recently some botnet are observed to use a new more difficult to detect protocol named as random protocol.

There are different verticals and mechanims to detect the families of botnet as it is always difficult to detect the advance botnet through single technique. Figure 4 shows the overall processing in the analysis framework. As earlier explained in the proposed framework Honeynets captures the data at system level and the network level. At System level data in the form of windows executable files and is captured by the file system integrity checker. Whereas at network level the network traffic in PCAP format is captured by the TCPDUMP tool. Both of these data feeds from Honeynets are processed separately by the separate analysis engines. The collective analysis result correlating the outcome of both analysis engines are generated by the analysis framework.

TABLE I. SOME BOTNETS AND THEIR PROTOCOLS

Botnet Families	Type of botnet
Agobot, SDBot, RxBot	IRC
Zeus, SpyEye,ZeroAccess	HTTP
EggDrop,Phatbot, Ramnit	P2P

Windows Executable processing: windows executable obtained from file system integrity checker and PCAP data through Portable executable extractor code are first processed through Antivirus engines at Virustotal.com [11]. Antivirus engine labels them into three classes **1) Bot, 2) Non-Bot and 3) Not-detected** malwares. The bot class of malwares are further mapped with the network PCAP data with the help of timestamp. The contextual information of bot binariers along with the network logs are processed through the analysis engines. The class "Non-Bot" malwares are not considered assuming they are not possessng any botnet infection traces.The malware samples which are not given any lable, are further executed in the sandbox enviornment for the dynamic analysis as these executables could be undetected bot samples. In parallel, the network data(PCAP) corresponding to "Not Detected" class of executables are processed through the botnet analysis engines to get botnet attributes. The algorithm for processing windows executable samples obtained from Honeynet is as follows:

Start {

1. Extract the malware samples from honeypot machine.

a. The binary samples dropped on a honeypot OS is captured and logged into a specific directory.

2. Extract the Windows executable from PCAP data using Portable Executable extractor tool.
3. Label malware samples through AV scanner
   a. If Labelled by any AV as Bots
      i. Map malware with network logs(PCAP)
      ii. Extract the botnet attributes from corresponding PCAP
   b. Else
      i. Execute the malware in sandbox environment and analyze the system changes

Network Traffic Processing: PCAP data from honeynet is submitted to the analysis engnines which comprises of

1)Bothunter[12],
2)Automated Payload Inspections(API),
3)Snort based detection engine to extract the botnet attributes (i.e C&C IP, Egg download IP, Domain names etc).

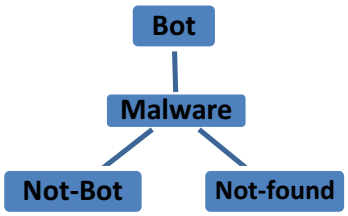

Fig. 4. Malware classes

The same PCAP data is then processd through the windows Portable executable file extractor. This module extracts the windows executable for the PCAP data and then submits it to the Antivirus engine for labeling of malware samples.The algorithum for processing of the PCAP data is as follows:

1. Process the PCAP data through analysis framework
   c. If botnet communication found
      i. Extract the botnet attributes
      ii. Correlate the analysis results of multiple engines
      iii. Bot infection determinations
   d. Else
      i. Manual Inspection of network logs
      ii. Packet checking and DNS extraction
      iii. DNS check with reputation engine

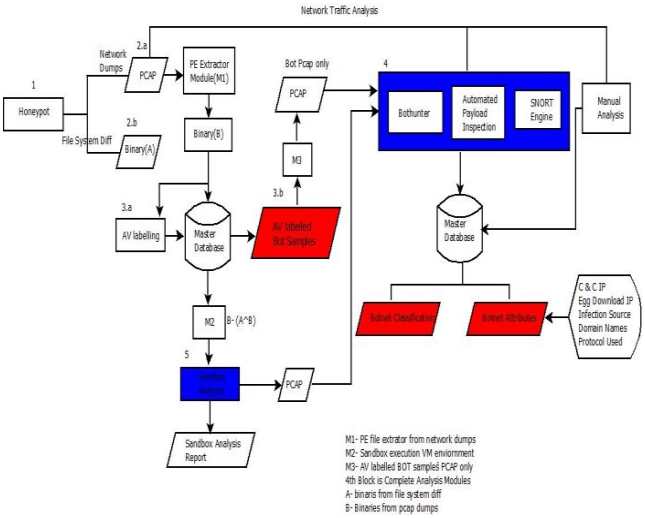

Fig. 5. Botnet Detection Framework

V. Experiment Results

As discussed above, two classes of malwares were analyzed through the analysis framework: "Bot" and "Not-Detected" as per the AV scanners.

A. Analysis of Known Bot Class

Table 2 shows the collection statistics for honeypots types with the Botnet families captured by them.

TABLE II. BOTNET METADATA WITH HONEYPOT OS

Botnet Name	Honeypot OS	Count
WormWin32/Dorkbot.I	XPSP3, XPSP1	65
W32/Baxbo.A	Nepenthes	3
WormWin32/Ramnit.A	XPSP2	4

B. AV Labeling of Known Bot Class

As the first step in the analysis process of the windows executable is Antivirus labeling Table 3 shows the labeled assigned to the captured bot families by popular antivirus engines.

TABLE III. AV LABELING OF KNOWN BOT

Botnet name	Kaspersky	Avast	McAfee	Microsoft	Symantec
WormWin32/Dorkbot.I	Trojan-Dropper.Win32.Dapato.azdf	Win32 FakeAlert-CKQ [Trj]	Worm Win32/Dorkbot.I	Worm Win32/Dorkbot.I	Not-labeled
W32/Baxbo.A	Net-Worm.Win32.Bobic.bc	Win32 VanBot-DB	W32/Bobax.worm.gen	Virus Win32/Bobax.A	Not-labeled
WormWin32/Ramnit.A	Not labeled	Win32/Ramnit.A	Not-labeled	Win32/Ramnit.A	

The detailed analysis of all the three botnet families is given below. We have presented the results of botnet family along with the analysis using automated botnet detection tools (Bothunter) and in-depth traffic analysis.

C. Detailed Analysis of Dorkbot

Dorkbot gained publicity in late 2011 for an attack on Facebook's chat system, with users receiving a message with a bogus link that appeared to come from one of their Facebook friends. A similar Dorkbot worm appeared later in the same year, this time preying on Twitter users [14].

DorkBot is an advance IRC Bot sharing similarity with NgrBot. That's why most of the antivirus software labels DorkBot samples as NgrBot. Our botnet monitoring system has captured the samples of both NgrBot and DorkBot. During detailed analysis of the samples it was observed that both of them have some differences in their characteristics [12].

DorkBot typically spreads through infection vectors such as instant messaging, USB removable drives, websites or social media channels like Facebook and Twitter. After getting in to the victim's machine it opens a backdoor on infected computer, allowing remote access and making the system a Bot.

TABLE IV. BOTNET ATTRIBUTES OF DORKBOT

Dorkbot	Values
C & C IP / Domain	113.75.x.x
Server port	6060
Client port	Random
Egg Download	113.106.x.x,60.165.x.x, 118.212.x.x

The botnet attributes are extracted from the network traffic mapped for the labeled bot samples. Table 4 shows the attributes extracted for DorkBot.

1) Data Extracted from honeypot sensor

65 samples of the Dorkbot were captured by the windows XP (SP1) based high interaction honeypot.

Results: Binary Sample MD5

85ae6abc0714f070cdf70335d752fc75,
d271d8ba5c6f9eeb887d45c5a2a8abd2,
e13f02acc31b3d0606f89972f9210418 etc.

2) Bothunter Analysis results

When the PCAP data of the Honeypot was processed by the Bothunter. The presence of Bot infection cycle were confirmed and following botnet attributes were extracted.

Results: Botnet attribute

C & C IP Address: 113.75.x.x

Egg download: 113.106.x.x, 60.165.x.x, 118.212.130.x

++++++++++++++++++++++++++++++++++++

EGG DOWNLOAD:

113.106.x.x (4) (04:06:50.230 IST-04:07:23.584 IST)\

event=1:3300001 {tcp} E3[rb] BotHunter Scrip-based Windows egg download .exe, [] MAC_Src: 08:00:27:61:EE:1A\

1082->6060 (04:06:50.230 IST)\

C&C TRAFFIC:

113.75.x.x (17) (04:14:21.419 IST)\

event=1:2010859 (9) {tcp} E4[rb] ET TROJAN Gh0st Trojan CnC, [] MAC_Src: 08:00:27:61:EE:1A\

1083->8000 (04:14:21.419 IST)\
1084->8000 (04:15:13.103 IST)\
1085->8000 (04:16:03.904 IST)\
1086->8000 (04:16:55.640 IST)\
1087->8000 (04:17:46.528 IST)\
1088->8000 (04:18:38.343 IST)\
1089->8000 (04:19:29.038 IST)\
1091->8000 (04:23:05.003 IST)\
1092->8000 (04:23:56.780 IST)\

----------\

event=1:2016922 (8) {tcp} E4[rb] ET TROJAN Backdoor family PCRat/Gh0st CnC traffic, [] MAC_Src: 08:00:27:61:EE:1A\

++++++++++++++++++++++++++++++++++++

EGG DOWNLOAD\ 60.165.x.x (05:17:37.166 IST)\

event=1:3300001 {tcp} E3[rb] BotHunter Scrip-based Windows egg download .exe, [] MAC_Src: 08:00:27:61:EE:1A\

1137->80 (05:17:37.166 IST)\

C&C TRAFFIC\

113.75.x.x (17) (05:10:21.337 IST)\

event=1:2010859 (9) {tcp} E4[rb] ET TROJAN Gh0st Trojan CnC, [] MAC_Src: 08:00:27:61:EE:1A\

1129->8000 (05:10:21.337 IST)\
1130->8000 (05:11:20.102 IST)\
1131->8000 (05:12:10.906 IST)\
1132->8000 (05:13:02.559 IST)\
1133->8000 (05:13:53.449 IST)\
1134->8000 (05:14:44.952 IST)\
1153->8000 (05:24:02.380 IST)\
1162->8000 (05:26:57.353 IST)\
1168->8000 (05:28:49.839 IST)\

----------\

event=1:2016922 (8) {tcp} E4[rb] ET TROJAN Backdoor family PCRat/Gh0st CnC traffic, [] MAC_Src: 08:00:27:61:EE:1A\

1129->8000 (05:10:21.337 IST)\
1130->8000 (05:11:20.102 IST)\
1131->8000 (05:12:10.906 IST)\
1132->8000 (05:13:02.559 IST)\
1133->8000 (05:13:53.449 IST)\
1134->8000 (05:14:44.952 IST)\
1153->8000 (05:24:02.380 IST)\
1162->8000 (05:26:57.353 IST)\

3) Automated payload based inspection

A total of 5 binaries were logged in payload result file, these are: aa.exe, ddn.exe, rse.exe, bao.exe, 360sd.exe. During payload inspection through offline analysis, the ddn.exe the communication with IP: "069.197.1x.x" which a likely Command and Control IP. Also, the suspicious DNS query information to "api.wipimania.com" is observed.

D. Detailed Analysis of W32/ Baxbo.A

Table 5 shows the details of the botnet attributes of W32/Baxbo.A botnet family. This botnet was captured at the low interaction Honeypot. The further detailed analysis of this botnet family is given below.

TABLE V. BOTNET ATTRIBUTES OF BOBAX

W32/Bobax.A	Domain
C & C IP / Domain	yutunrz.1dumb.com, jdjsloy.dynserv.com
Server port	80
Client port	Random

1) Analysis through Bothunter:

Result: No bot infection cycle traces were detected by Bothunter.

2) Automated Payload Inspection

Result: Communications were observed with the following Domains was observed

TABLE VI. W32/BOBAX.A DOMAINS COMMUNICATION

ftp.icq.com
yutunrz.1dumb.com
xx.enterhere.biz
storage.ovip.icq.com
mx1.hotmail.com
ftp.icq.com
mailin-02.mx.aol.com
yutunrz.1dumb.com
mcduii.3-a.net
jdjsloy.dynserv.com
wyqggvow.afraid.org
wyqggvow.afraid.org
xx.ka3ek.com
xx.enterhere.biz

From the above mentioned domain names, the domain names yutunrz.1dumb.com, jdjsloy.dynserv.com are C&C domains and are listed by botnet tracking engines and reputation engine as malicious [15].

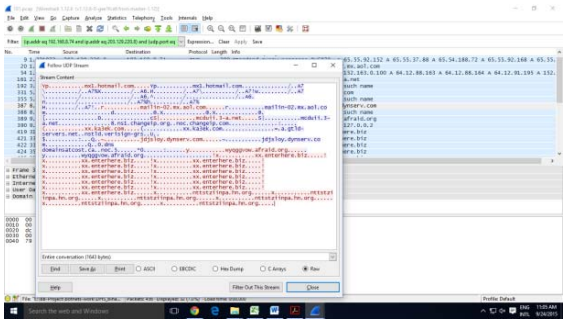

Fig. 6. Snapshots of C&C communication traces

3) Sandbox Analysis

Result: Table 7 shows the results of sandbox execution of the W32/Baxbo.A malware sample.

TABLE VII. OBSERVED SANDBOX BEHAVIOR

Alters Windows Firewall	✓
Copies to Windows	✓
Checks For debugger	
Could Not Load	
Creates DLL in System	
Create EXE in System	
Create Hidden File	
Create Mutex	✓
Create Service	
Delete File in System	
Deletes Original Sample	✓
Hooks Keyboard	
Injected Code	
Make Network Connection	✓
Modifies File in System	
Modifies Local DNS	
Opens Physical Memory	
Start EXE in Documents	✓
Starts EXE in Recycle	
Start EXE in System	✓
Windows/Run Registry Key Set	
More than 5 process	✓

E. Detailed analysis of WormWin32/Ramnit.A

Table 8 shows the botnet attributes extracted for the Ramnit botnet class.

TABLE VIII. BOTNET ATTRIBUTES OF RAMNIT

WormWin32/Ramnit.A	Domains
C & C IP / Domain	zahlung.name, glavdmn.com, fget-career.com
Server port	80
Client port	Random

4) Analysis through Bothunter

Result: No specific results as bot-profile score is not generated

1) Automated payload inspections

Result: Figure 8 shows the result of the automated payload inspection module. It shows the list of the Domain names generated by the malware and the IP addresses in which the domain names got resolved.

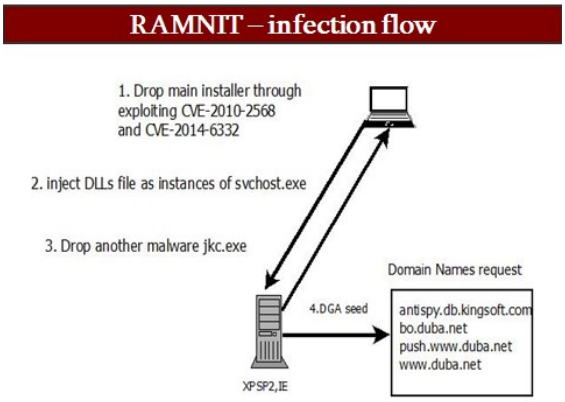

Fig. 7. Ramnit botnet structure

DOMAIN NAMES (124 domains)

Sr. No.	Domain Name	IP address
1	conf.f.360.cn	106.120.167.29 106.120.167.13 106.120.167.16 106.120.167.13
2	qianchuo.3322.org	219.238.115.124
3	qup.f.360.cn	106.120.167.29 106.120.167.13
4	u.qurl.f.360.cn	106.38.187.108 106.120.167.92
5	qurl.f.360.cn	106.38.187.108 106.120.167.92
6	qurl.qh-lb.com	106.38.187.108 106.120.167.92
7	sdup.360.cn	205.251.251.23 205.251.251.98 205.251.251.51 205.251.251.45 205.251.251.229 205.251.251.187

Fig. 8. Domain names observed for Ramnit.A

2) Manual Analysis

Result: Manual Analysis performed through reputations

- **Domains resolved** legitfreecounters.com, forever-counters.com, google-updaete.com
- **IP Recorded in communications :** 213.108.x.x, 66.85.x.x
- **IP Address used to phone hone during installation :** 213.108.x.x

F. Analysis of unknown not-labeled class of malware

The following sets of binaries are extracted from distributed honeynet data sets and were not labeled by any of the Antivirus scanners. But when investigated through Bothunter and our automated payload inspection modules, traces of botnet communications and were observed.

TABLE IX. UNLABELED BINARY AS VIRUSTOTAL

Binary MD5	Binary ID	VT Results
14109e9b4d670bece54c5 e94593d54e9	B4491.exe	Not in database
1f435df4be46abf9181e14 fc59f2571f	B4494.exe	Not in database
2754d088e7fb25879c3b9 3c0f0503cbf	B4504.exe	Not in database
70c537d6db3580acd641b 6b6e4c24ed6	B4510.exe	Not in database
b8726601a1c4053b01be1 49fd719022b	B4517.exe	Not in database
554c11ba02f7547308568 8d6762444b7	B4539.exe	Not in database
5a6c80490a4e9683fa9fdc e1d1b92c72	B4905.exe	Not in database
555165cf26adc0b97bb9b cd79991ad85	B4900.exe	Not in database
ce8b3ff04854c63d7c61ef 1a370ed216	B4581.exe	Not in database

Table 9 shows the list of the malware samples which were not detected by any antivirus engine but have shown bot characteristics when analyzed. This proves that Honeynets are capable of detecting the zero day malware samples. Table 20 shows the MD5 of the windows executable which haven't shown any malicious behavior. These samples could be the windows updates or possible malicious codes with ability to evad out detection mechanism.

TABLE X. UNLABELED (W.R.T VIRUSTOTAL) BINARY WITH BOT BEHAVIOR

d05c493f2b49d7cd854f49850b757b0b	B4653.exe
e54d83be6da7f5c7013e281de0ccad3a	B4722.exe
0462ec75ea788233beab52930ef007bb	B4724.exe
9913b0c8ebe097ce007791c32036ec43	B4800.exe
1ba97c949813c7eddc4cce379c37e546	B4802.exe
4758ad18f0d1c57ad12f856c545eca67	B4806.exe
d05c493f2b49d7cd854f49850b757b0b	B4653.exe
e54d83be6da7f5c7013e281de0ccad3a	B4722.exe

TABLE XI. BOTNET ATTRIBUTES OF UNLABELED BINARIES

BinID	C & C IP	Egg Download IP	Infection source
B4539.exe	-	210.14.x.x 113.10.x.x 122.226.x.x	-
B4491.exe	-	198.2.x.x	-
B4517.exe	-	210.14.x.x 117.135.x.x	-
B4494.exe	-	210.14.x.x 122.226.x.x 210.14.x.x 203.86.3.x.x	-

B4504.exe	-	117.135.x.x 122.226.x.x	-
B4510.exe	-	117.135.x.x	-
B4932.exe	-	-	113.31.x.x 122.226.x.x 114.36.x.x
B4905.exe	-	108.171.x.x 58.221.x.x 184.168.x.x	109.230.x.x 180.153.x.x 109.230.x.x 74.208.x.x 180.153.x.x 109.230.x.x 103.7.x.x 103.7.x.x
B4900.exe	-	223.4.212.x.x 108.171.x.x	-
B4581.exe	206.25 1.x.x 60.173. x.x 206.25 1.x.x	124.173.x.x	-

Table 11 shows the botnet attributes (C&C IP, egg download source, attacking IP etc.).

VI. CONCLUTION AND FUTURE WORK

Here in this paper we have presented a framework for botnet detection. We also presented experimental results and detailed case studies of few botnet families. The shortcomings and the limitations of popular botnet detection technique were also highlighted.

Based upon the experimental results it could be concluded that no single tool or approach can detect all botnet classes and hence there is a need for a unified framework; a combination of multiple approaches for the botnet detection.

In the future work we would like to address the problem of labeling the unlabeled class of bot binaries using protocol analysis with machine learning.

Acknowledgment

We would like to thank Director, CDAC-Mohali for his continuous encouragement. We also would like to thank the entire Cyber Security Technology Division (CSTD), CDAC-Mohali for their cooperation and support.

REFERENCES

[1] J.S.Bhatia, Rakesh Sehgal and Sanjeev Kumar, "Botnet Command Detection using Virtual Honeynet", International Journal of Network Security & Its Applications (IJNSA), Vol.3, No.5, Se p 2011,pp. 177-189 DOI :10.5121/ijnsa.2011.3514

[2] Sanjeev Kumar, Rakesh Sehgal and J.S. Bhatia, "Hybrid Honeypot Framework for Malware Collection and Analysis", 7th International Conference on Industrial and Information Systems (ICIIS-2012), August 6-9, 2012, IIT Chennai, Published in IEEE Xplore.

[3] Sanjeev Kumar, Paramdeep Singh, Rakesh Sehgal, and J.S.Bhatia, "Distributed Honeynet System using Gen III Virtual Honeynet" International Journal of Computer Theory and Engineering, Vol. 4, No. 4, August 2012, IJCTE 2012 Vol.4(4): 537-541 ISSN: 1793-821X

[4] Paul Bacher, Thorsten Holz, Markus Kotter, and Georg Wicherski. Know your Enemy: TrackingBotnets. http://www:honeynet:org/papers/bots/, 2007.

[5] Pedro Correia et.al, Statistical Characterization of the Botnets C&C Traffic, INSODE 2011

[6] http://www.edgissecurity.org/honeypot/dionaea/

[7] http://www.tcpdump.org/

[8] Gautam, R., Kumar, & Bhattacharya, J. "Optimized Virtual Honeywall with Implementation of Host machine as Honeywall" 12th IEEE India International Conference (Accepted)

[9] http://ipset.netfilter.org/iptables.man.htm

[10] https://www.snort.org/

[11] https://www.virustotal.com/

[12] Guofei Gu et.al, BotHunter: Detecting Malware Infection Through IDS-Driven Dialog Correlation,

[13] https://blog.fortinet.com/post/dorkbot-a-twin-botnet-of-ngrbot

[14] http://www.webopedia.com/TERM/D/dorkbot.html

[15] Paul Royal, On the Kraken and Bobax Botnets, April,2009, Damballa

[16] https://www.r-project.org/

[17] Moheeb Abu Rajab Jay Zarfoss Fabian Monrose Andreas Terzis.: A Multifaceted Approach to Understanding the Botnet Phenomenon. In: 6th ACM SIGCOMM conference on internet measurement ACM 2006

[18] Elizabeth Stinson and John C. Mitchell Department.:Characterizing Bots' Remote Control Behavior.In: International conference on detection of intrusion & malware and vulnerability assessment,2007

[19] Guofei Gu, Phillip Porras, Vinod Yegneswaran, Martin Fong, Wenke Lee" BotHunter.: Detecting Malware Infection through IDS-Driven Dialog Correlation.In: 16TH USENIX SECURITY SYMPOSIUM 2008

[20] 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

[21] Brumley, D., Hartwig, C., Liang, Z., Newsome, J., Poosankam, P., Song, D., and Yin, H. 2007: Automatically identifying trigger-based behavior in malware.In: Book chapter in Botnet Analysis and Defense

[22] Saurabh Chamotra, Rakesh Kumar Sehgal, Raj Kamal.: HoneySand: An open source tools based sandbox environment for Bot analysis and Botnet Tracking. In: Special issue of International Journal of Computer Applications on Communication Security, No.7, /mar 2012

[23] Rakesh Kumar Sehgal, D.B. Bhilare, Saurabh Chamotra.: An Integrated Framework for Malware Collection and Analysis for Botnet Tracking.In: Special Issue of International Journal Computer Applications (0975-8887) on Communication Security, No. 10, Mar 20122

[24] D. Dagon, G. Gu, C. Lee, and W. Lee.:Taxonomy of Bot-net Structures. In: Annual Computer Security Applications Conf. (ACSAC), 2007.

[25] G. Gu, V. Yegneswaran, P. Porras, J. Stoll, and W. Lee. " Active Botnet probing to identify obscure command and control channels". In Proceeding of Annual computer security application conferences, asac, pp.241-253, 2009.-Botprobe[

[26] Guofei Gu, Junjie Zhang, and Wenke Lee. "BotSniffer: Detecting Botnet Command and Control Channels in Network Traffic." In Proceedings of the 15th Annual Network and Distributed System Security Symposium (NDSS'08), San Diego, CA, Febrnary2008

[27] G. Gu, R. Perdisci, J. Zhang, and W. Lee, "Botminer: Clustering analysis of network traffic for protocol- and structure independent Botnet detection," in Proc. 17th USENIX Security Symposium, 2008

[28] J. Kinder, S. Katzenbeisser, C. Schallhart, and H. Veith: Detecting Malicious Code by ModelChecking. In:Conference on Detection of Intrusions and Malware & Vulnerability Assessment (DIMVA), 2005.

[29] Christodorescu, M., Jha, S: "Static analysis of executable to detect malicious patterns.In: Proceedings of the 12th USENIX Security Symposium, 2003

[30] H. Crawford and J. Aycock. Kwyjibo: Automatic Domain Name Generation. In Software Practice and Experience, John Wiley & Sons, Ltd., 2008