DESIGN OF A HONEYPOT BASED NETWORK DEFENSE SYSTEM TO COUNTERATTACK MALWARES AND BOTNET

A FRAMEWORK DESIGN

[1]Suhel Ahamed
[1]Assistant Professor
Department of Information Technology, Institute of Technology
Guru Ghasidas Vishwavidyalya (Central University)Bilaspur, India
Email:[1]suhel.yusuf@gmail.com

Abstract— Today the world requires secure and trust based computing, wherein Malwares (Virus, Worms, Trojans, etc.) and intrusions are continues threat to trust based computing. Nowadays script kiddies with less knowledge are more malicious then the expert malware writer. They cause the outbreak of several digital disasters without control. Also the botnet plays important role to propagate malwares and support anonymous intrusion. The proposed solution targets these script kiddies by using social engineering on a honey pot based system. The proposed system counterattacks the remote machines on botnet by a benevolent looking bait to gain control and to break the propagation chain up to a certain extent. In this paper a framework design is presented for the proposed solution.

Keywords— Honeypot; Social Engineering; Network Defense System; Script kiddies; Botnet; Rootkit; Trojans; Virus; Worms.

I. INTRODUCTION

From first large-scale computer virus outbreak in history to recent threats like *heart-bleed*, Malwares and intrusions proves to be a serious threat to trust based computing. In the past few decades there where several new types of malwares found in the wild (IBM researcher David Chess, coined the term in the wild to describe computer virus that was encountered on public network and production systems [4]). Malware writing also evolved through new technologies from simple file infectors or cloner to polymorphic, metamorphic, and retro virus. Several malwares Use packing, obfuscation to avoid detection; whereas rootkits, function hooking is used to gain control of the machine and to convert it to Bots. Where polymorphic virus use mutation engine to generate a new variant or mutant of the virus to avoid detection; retro type virus and malwares use the techniques to destruct the computer / network defense system and antivirus programs.

The malware writers and intruders also use Social Engineering to gain access and propagate through networks. Social engineering is not a technical attack; it does not exploit vulnerability in a program. Instead, it is a psychological attack which exploits vulnerabilities of the user. Intruders build up trust of the user, pretending to be a person or organization the user knows. They then exploit this trust by obtaining access to the computer and/or passwords of user. These attacks are launched using the same tools the end user uses every day, such as email, phone, and web [2], e.g. the malwares such as Simpsons.Trojan [8], and W32.Fujacks [7] uses this technique that

exploits the sentiments of end user to create a profit chain.

Botnets, networks of malware-infected machines that are controlled by an adversary, are the root cause of a large number of security problems on the Internet. Bot is a type of malware that is written with the intent of taking over a large number of hosts on the Internet. Once infected with a bot, the victim host will join a botnet, which is a network of compromised machines that are under the control of a malicious entity, typically referred to as the Bot-master. Botnets are the primary means for cyber-criminals to carry out their nefarious tasks, such as sending spam mails, launching denial-of-service attacks, or stealing personal data such as mail accounts or bank credentials. This reflects the shift from an environment in which malware was developed for fun, to the current situation, where malware is spread for financial profit [1].

In the past few years, virus writers have become more cautious, and craftier. These days, many elite Malware writers do not spread their works at all. Instead, they "publish" them, by posting their code on Web sites, often with detailed descriptions of how the program works. Essentially, they leave their virus lying around for anyone to use. The people who release the virus are often anonymous mischief-makers, or "script kiddies." That's a derived term for aspiring young hackers, usually teenagers or curious college students, who don't yet have the skill to program computers but like to pretend they do. They download the virus, claim to have written them themselves and then set them free in an attempt to assume the role of a fearsome digital menace. Script kiddies often have only a dim idea of how the code works and little concern for how a digital plague can rage out of control [6].

In this paper a solution is proposed to deal with these problems. A framework design is presented here for a honeypot based network defense system that uses social engineering against these script kiddies and tries to gain control over the botnet machines to break their chain of propagation up to a certain extent thus preventing the digital disaster at the network level and reducing the effect of overall damage..

II. HONEYPOT

A. *Definition*

A honeypot is a deception trap, designed to entice an attacker into attempting to compromise the information systems in an organisation. If deployed correctly, a honeypot can serve as an early-warning and advanced security surveillance tool, minimising the risks from attacks on IT systems and networks. Honeypots can also analyse the ways in which attackers try to compromise an information system, providing valuable insight into potential system loopholes[3].

Fred Cohen introduced the first publically available honeypot solution, the deception toolkit in 1997[4]. Honey pots can be a computer or a network of computer, or a service or can be a simple file with some valuable looking data that attracts the attacker to compromise it. Honeypots are generally classified in to two types, low interaction honeypot and high interaction honeypot. A low interaction honeypot generally emulates any network service or a simple file; whereas a High interaction honeypot presents itself as a vulnerable real computer with an operating system or a network of computer that may include some real computer and some virtual machines.

B. *Previous Work*

Some important honeypots are mentioned here for the purpose of this paper.

Honeyd [9] by Niels Provos it is a small daemon that creates virtual hosts on a network. The hosts can be configured to run arbitrary services, and their personality can be adapted so that they appear to be running certain operating systems. Honeyd enables a single host to claim multiple addresses.

HIHAT [10] The High Interaction Honeypot Analysis Toolkit (HIHAT) allows to transform arbitrary PHP applications into web-based high-interaction Honeypots. Furthermore a graphical user interface is provided which supports the process of monitoring the Honeypot and analyzing the acquired data.

LaBrea [11] also called sticky honeypot developed by Tom Liston can capture ARP request on your network, very effectively slowing down worms on a network. LaBrea takes over unused IP addresses, and creates virtual servers that are attractive to worms, hackers, and other denizens of the Internet. The

program answers connection attempts in such a way that the machine at the other end gets "stuck", sometimes for a very long time.

C. *Why Honeypot for this project?*

Honeypots have the capability of attracting the intruders by using Social Engineering against them. For the purpose of this project a new type of honeypot is required which is highly interactive with the remote machines or bots and which is also capable of enticing the intruder to self-inject several programs in to the veins of Botnet to prevent the propagation chain up to a large extent, thus increasing the security in trust based computing.

III. FRAMEWORK DESIGN

The Framework design is presented here for the proposed Honeypot based Network Defense System. As illustrated in the block diagram Fig.1 the honeypot based network defense System has three main components:

A. *A new type of Honeypot*

Traditional honeypots does not fulfill the requirement of this project as it is only used to record the compromising behavior of the intruder or malware, so a new type of Honeypot needs to be conceptualized.

- This honeypot is capable of attracting the intruders and malwares by the use of social engineering.
- It should create valuable looking resources on real or virtual computers sufficient enough to lure the intruder and malware to compromise it.

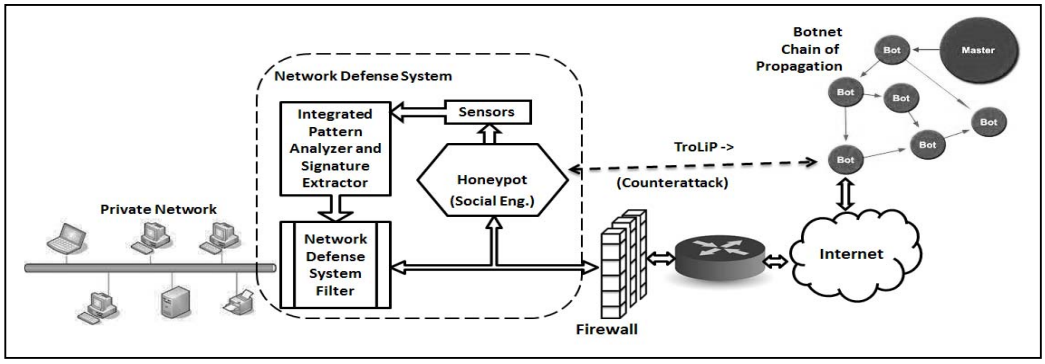

Fig. 1. Honeypot Based Network Defense System Counterattacking Botnet chain of Propagation.

- It should also contain some benevolent looking files with Trojan like program (TroLiP) hidden inside it. So when the intruder tries to steal it using a botnet and copies the file to the nearest bot, it actually takes a Trojan horse home and unknowingly activates it. The Trojan like program inside it then opens up some unused ports and downloads other supporting programs.
- It will also use rootkits and function hooking methods to gain administrative control over that bot. after that it will start cleaning the bot for its entire malcontents.
- The Trojan like program will repeat the process to all those bots who has copied it, to transfer the valuable looking file to the master node.

In this way it will be successful in counterattacking the intruder or malware up to a large extent depending upon the design of its distribution channel.

B. *Integrated pattern analyzer and Signature extractor*

The honeypot collects necessary information, traces, Behavioral Patterns and samples of malwares through attached sensors and monitors; they are then analyzed by the integrated pattern analyzer and Signature extractor. It is based on the concept discussed by Suhel Ahamed et al. [5].As illustrated in Fig.2. Static signature S(sig) obtained by static analysis of suspicious malware samples and Dynamic signature D(sig) obtained by dynamic analysis of patterns, traces and behavioral data of intruder or malware an integrated stronger signature I(sig) can be formulated as:

$$\text{Integrated Signature } I(sig) = \int (S(sig) + D(sig))$$

Where $\int()$ is the integration function to create two layer framed and concise signature I(sig) which can be stored in a Signature (DB)

Database for scanning and identification purpose in this process.

C. *Network Defense System Filter*

The third component of Network Defense System is a special filter which is a Bastion Host used to protect the private network from outside attacks by intruders and malwares.

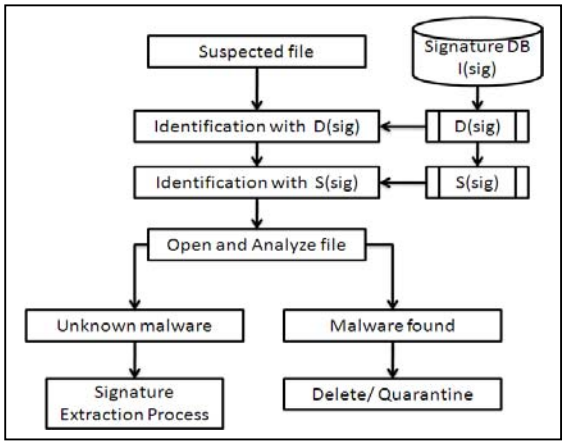

Fig. 2. Analysis for malwares and signature extraction. [5]

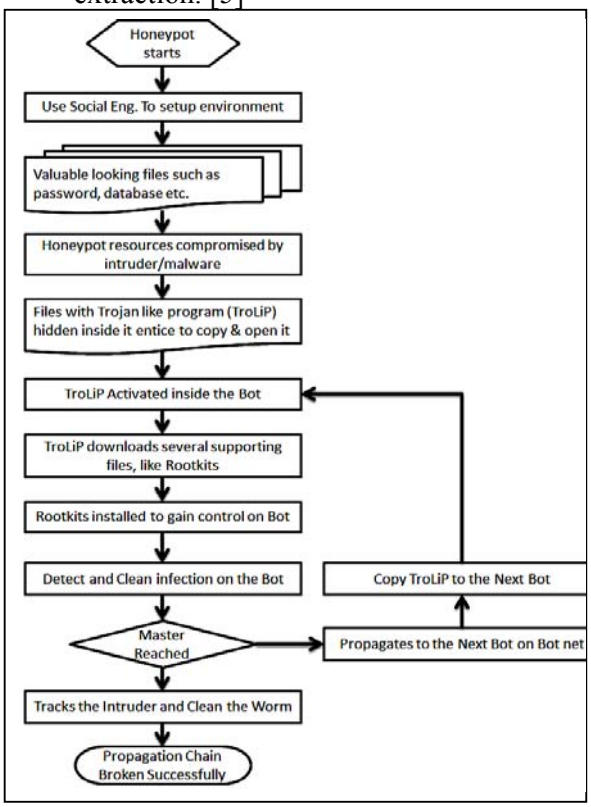

Fig. 3. Framework Process Flow

The network Defense System has a two way action plan:

1) Protecting Private Network: The internal Private Network should always be protected so it uses cleaning and filtering Techniques for inbound traffic by using the Signatures and Knowledge received from integrated pattern analyzer and signature extractor. Filter also continuously monitors the inbound and outbound traffic and collects the suspicious files for analysis and signature extraction.

2) Counterattack: On the other direction the honeypot based system entice the intruder or malware to steal some valuable looking files and copy to the nearest bot chaining up to the master. Where the Trojan like program (TroLiP) hidden inside it gets activated and starts the cleaning process of bot and breaks the propagation chain of malware and botnet, thus reducing traffic load increased by attacking and slowing down the malware or intruder.

IV. SCOPE AND LIMITATION

A. *Scope*

Security for Trust based Computing is very important. As discussed in this paper most of the virus and intrusion attempts are done by the Script Kiddies (defined earlier in this paper) who have less knowledge and concern about the malware and its use and are less worried about the aftermath of the Cyber Crime they have done. Also they are less careful for their operation and very reluctant about the resources they use; such as they do not care about the bots they have used to propagate anonymously through internet, and they also don't expect the similar counterattack from the target to the Bot. this vulnerability of the attacker can be used against him.

This motivates the design of a new type of Honeypot which can use Trojan like Programs (TroLiP) to exploit the vulnerability of the intruder or malware.

B. *Limitation*

This design has some limitation with the expert malware writers and smart intruders.

1) The smart intruders may sense the presence of a TroLiP and can stop or delete it. Furthermore he can analyze the TroLiP and can create a vaccination for it.

2) The malware writers can also analyze the TroLiP and can equip their malwares to stop or delete the TroLiP.

V. DEVELOPMENT AND EXPERIMENTATION

The author of this paper is very much interested in Development of the proposed Honeypot based Network Defense System, and to perform experimentation to prove the effectiveness of the Solution. A real computer network scenario and real computers with

capability of virtualization are required for the experimentation work. The Development and Experimentation should fulfill certain requirements:

1) The solution should be platform independent and required to be developed on such environment and programming languages.

2) Core machine level architecture should be considered for time and space complexity.

3) The network communication parameters should be considered and therefore the components such as TroLiP and other supporting files should be light weight.

4) The TroLiP should use the stealth techniques to hide itself from detection.

5) The design should have multiple points of operation i.e. the components should be separated on multiple points on real machines.

6) The Experimentation should be done on a secure and emulated environment. Also the monitors should save the output data on a separate machine for securing the experimentation results.

7) The experimental network should not be connected to internet. It should use an experimental internet in a box facility created for the sole purpose of this project.

8) The malware collected and used for the purpose of experimentation should be in tight captivity and should not be let loose in outer world.

9) All the experiments and developments should follow the national and international legal policies.

VI. FUTURE WORK AND CONCLUSION

In consideration for the limitations discused in this paper The future work of the project would consider the development of smart (TroLiP) and related programs that can use stealth techniques to hide itself and subvert detection in the remote bot machine.

The development of new security methods and technologies is very important for the trust based computing, and to boost up the trust of end user to use internet facilities like e-commerce and m-commerce. This paper presents idea and design of a honeypot based network system to counterattack the malwares and intruder and breaks their chain of propagation up to a certain extent. This also requires the development of a new type of honeypot to fulfill the purpose of this project.

REFERENCES

[1] Brett Stone-Gross et.al., "Your Botnet is My Botnet: Analysis of a Botnet Takeover", CCS'09 November 9–13, 2009, ACM.

[2] "Cyber Security Newsletter", The SANS Institute, 2012, University of Alabama, Birmingham, www.uab.edu/it/security, retrived on 20th april 2014

[3] "Honeypot Security", february 2008. The Government of the Hong Kong Special Administrative Region, retrived on June 2013

[4] Peter Szor, " The Art of computer Virus Research and Defense", Symantec Press, Addison Wesley Professional, 2005.

[5] Suhel Ahamed, Dr. J.L. Rana, R.K. Pateriya, "Integrated Approach for Signature Extraction and Profile Generation of Malwares with Monitoring and Detection", Proceedings of ICCNS 08 , 27-28 September 2008, pp.30-34

[6] Clive Thompson, "The Virus Underground", The New York Times, 2004, www. nytimes.com, retrived on June 2013

[7] Robert X Wang, "The Panda Outlaw:W32.Fujacks", White Paper, Symantec Security Response, 2007, www.symantec.com, retrived January 2008

[8] "Symantec Newsletter july 2000", Symantec Antivirus Research Center, http://www.symantec.com/avcenter/reference/newsletter/Jul00inews.html retrived June 2013.

[9] Honeyd project, http://www.honeyd.org/ : Downloaded April 2014

[10] HiHat, http://hihat.sourceforge.net/: Downloaded April 2014

[11] LaBrea, http://labrea.sourceforge.net/: Downloaded April 2014