INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH & DEVELOPMENT

ISSN: 2278 – 0211

An Advanced Hybrid Peer-To-Peer Botnet

M. Muthu Kumar
Centre for Information Technology and Engineering
Manonmaniam Sundaranar University
Tamil Nadu, India

Mrs. M. Pasupathi
M.TECH
Centre for Information Technology and Engineering
Manonmaniam Sundaranar University
Tamil Nadu, India

Abstract:

Bot nets are an upcoming technology that can be used to detect and analyze network attacks. A Bot net is an apparently vulnerable system deployed to be hacked. An analysis of current Bot net approaches has been made and it has been evaluated in how far these approaches can contribute to the analyzation process. Some tests have shown that Bot nets are exposed to lots of known attacks and noise that hide the valuable information about new attacks and vulnerabilities.

Introduction

In the last several years, Internet malware attacks have evolved into better organized and more profit-centered endeavors Email spam, extortion through denial-of-service attacks and click fraud represent a few examples of this emerging trend. "Botnets" are a root cause of these problems A "botnet" consists of a network of compromised computers ("bots") **connected to the Internet that is** controlled by a remote attacker ("botmaster"). Since a botmaster could scatter attack tasks over **hundreds or even tens of thousands of computers distributed across the Internet, the enormous cumulative bandwidth and large number of attack sources make botnet-based attacks extremely dangerous and hard to defend against. entire botnet may be exposed once a C&C server in the botnet is hijacked or captured by defenders. As network security practitioners put more resources and effort into defending against botnet attacks, hackers will develop and deploy the next generation of bot nets with a different control architecture**

Existing System

- **Traditionally, information security has been purely defensive. Firewalls, Intrusion Detection Systems, encryption; all of these mechanisms are used defensively to protect one's resources.**
- **A variety of detection tools exist such as Intrusion Detection systems (IDS) and firewalls, but the main problem is that they only react on reconfigured and therefore known attacks.**
- **In an existing system that will produce only the simulation result.**
- **BOTMASTER does not know the hackers IP address.**
- **This system can only run on single system.**

Scope

- **The aim of the Project is listed below,**
- To know about the Hacker's Activities and Motivation.
- **To allo**w the Hacker's to hack the network and monitoring the hacker's activities.

Proposed System

The primary purpose of a Honey net is to gather information about threats that exist.

A Honey net is a type of honey pot. Specifically, it is a high-interaction honey pot designed to capture extensive information on threats.

High-interaction means a Honey net provides real systems, applications, and services for attackers to interact with. It is through this extensive interaction we gain information on threats, both external and internal to an organization.

Proposed system can note the IP address of Hackers and can identify what type of file they want to access and what password and key was given by hackers to access the file.

This system can produce the real time result.

We can run it on more than one system without changing, and can run in single system too.

Project Modules

Data Control

Data Control is the containment of activity. It is what mitigates risk. By risk, we mean there is always the potential of an attacker using a Botnet to attack or harm non-Botnet systems. We want to make every effort possible to ensure that once an attacker is within our Botnet, they cannot accidentally or purposefully harm other non-Botnet systems. This is more challenging then it seems. First, we have to allow the attackers some degree of freedom to act. The more activity we allow the attackers to perform, the more we can potentially learn about them. However, the more freedom you allow an attacker, the more risk there is they will circumvent Data Control and harm other non-Botnet systems.

The balance of how much freedom to give the attacker vs. how much you restrict their activity is a decision every organization has to make themselves. Each organization will have different requirements and risk thresholds. Second, we have to control the attacker's activity without them knowing their actions are being controlled. One of the best ways to approach Data Control is not to rely on a single mechanism with which to implement it.

Instead, implement Data Control using layers, such as counting outbound connections, intrusion prevention gateways, or bandwidth restrictions. The combination of several different mechanisms helps protect against a single point of failure, especially when dealing with new or unknown attacks. Also, Data Control should operate in a fail closed manner. This means if there is a failure in your mechanisms (a process dying, hard drive full, misconfigured rules) the Botnet architecture should block all outbound activity, as opposed to allowing it.

One thing to consider with Data Control, it can only minimize risk. We can never entirely eliminate the potential of an attacker using a Botnet to harm non-Botnet systems. Different technologies and approaches to Data Control have different levels of risk, but none eliminate risk entirely.

Data Capture

Data Capture is the monitoring and logging of all of the black hat's activities within the Botnet. It is this captured data that is then analyzed to learn the tools, tactics, and motives of members of the black hat community. The challenge is to capture as much data as possible, without the blackhat detecting the process. As with Data Control, one of the primary lessons learned for Data Capture has been the use of layers. It is critical to use multiple mechanisms for capturing activity. Not only does the combination of layers help piece together all of the attacker's actions, but it prevents having a single point of failure.

The more layers of information that are captured, at both the network and host level, the more that can be learned. One of the challenges with Data Capture is that a large portion of attacker activity happens over encrypted channels (such as IPSec, SSH, SSL, etc). Data Capture mechanisms must take encryption into consideration. Also, just as with Data Control, we have to minimize the ability of attackers to detect our capture mechanisms. This is done several ways. First, make as few modifications to the botnets as possible. The more modifications you make, the greater the chance of detection. Second it is best that captured data not be stored locally on the botnet themselves. Not only could this data be detected by attackers, but it could also be modified or deleted.

As such, captured data must be logged and stored on a separate, secured system. Just as with Data Control, there are no guarantees with Data Capture. Attackers may identify ways to detect Data Capture mechanisms, and develop methods to bypass or disable them.

Data Collection

Data Collection is a third requirement, but this only applies to organizations that have multiple Botnets in distributed environments. Many organizations will have only one single Botnet, so all they need to do is both Control and Capture data. However, organizations that have multiple Botnets logically or physically distributed around the world, such as the Botnet Research Alliance have to collect all of the captured data and store it in a central location.

This way the captured data can be combined, exponentially increasing its value. Refer to Figure 1 to see an example of how the Botnet Research Alliance has achieved this. The Data Collection requirement provides the secure means of centrally collecting all of the captured information from distributed Botnets.

The Bot net Project has created a document that defines these three requirements in greater detail. The purpose of the document is to give organizations the flexibility to build a Botnet tailored to their environment and goals. However, the document ensures that the Botnets are effectively and securely deployed, allowing different Botnets to interoperate. Any organization considering deploying their own Botnets are encouraged to follow these requirements.

Data Analysis

In all three above cases, there are two steps you can take to help mitigate these risks, human monitoring and customization. By human monitoring, we mean you have a trained professional monitoring and analyzing your Honey net in real time. Anytime you suspect an attacker has successfully gained (or attempting to gain) access to one of your honey pots (such as detection of outbound connections, frequent established inbound connections, increased inbound traffic, transfer of files, unusual system activity, etc) a security professional should be monitoring and analyzing all captured data. This helps prevent the risk of an attacker detecting or disabling a Honey net, and attempting to harm other non-Honey net systems.

By having a human analyzing Honey net activity, instead of just depending on automated techniques, you help protect yourself against new or unknown attacks or Honey net countermeasures. In a worse case scenario, you can always shut down the Honey net down if the attacker has exceeded your organizations threshold for risk. Second, customization is critical. However, the best tool is to have human beings involved in monitoring, analyzing, and reacting to Honey net activity.

Data Encryption / Decryption

The Blow fish involves replacing each letter of the alphabet with the letter standing k places further down the alphabet

.

Encryption

Blowfish is a Feistel network consisting of 16 rounds The input is a 64-bit data element, x.

Divide x into two 32-bit halves: xL, xR

For i = 1 to 16:

xL = xL XOR Pi

xR = F(xL) XOR xR

Swap xL and xR

Swap xL and xR (Undo the last swap.)

xR = xR XOR P17

xL = xL XOR P18

Recombine xL and xR

Function F (see Figure 2):

Divide xL into four eight-bit quarters: a, b, c, and d

F(xL) = ((S1,a + S2,b mod 232) XOR S3,c) + S4,d mod 232

Decryption

It is exactly the same as encryption, except that P1, P2,..., P18 are used in the reverse order.

This algorithm used to encrypt the all the data before going to send to the user.

Using the private key k it is decrypted on the end user side. The user who knows the private key can only decrypt the data.

Logs And Alert System

This provide alert message to the administrator. If hacker entered into the network then the firewall detecting that hacker and immediately it tell to the honey pot about the hacker. Suddenly the Honey Pot monitoring the hacker activities.

Before that it gives the alert message to the log/alert server where the administrator is sited. Then the administrator can watch the hacker`s motivation and activities.

Functional Requirements

Requirement Analysis is a software engineering task that bridges the gap between system level software allocation and software design.

The basic aim of this stage is to obtain a clear picture of the needs and requirements of the end-user and also the organization. Analysis involves interaction between the clients and the analysis. Usually analysts research a problem from any questions asked and reading existing documents. The analysts have to uncover the real needs of the user even if they don`t know them clearly.

This is essential to ensure that the final specifications are consistent.

The 5 efforts are :

- **Problems and recognition**
- **Evaluation and synthesis**
- **Modeling**
- **Specification**

Conclusion

Honey nets can be a valuable addition to a security system.

Honey nets serve the community through their ability to collect and record information on black-hat activities.

A honey pot`s strength is its ability to divert an attacker from the main production system.

Once diverted, the attack can be studied and a countermeasure can be developed.

Reference

1. MathworksInc.:Simulink.www.mathworks.com/products/simulink.
2. Honeyd security advisory 2004-001: Remote detection via simple probe packet. http://www.honeyd.org/adv.
3. K. Anagnostakis,S. Sidiroglou, P. Akritidis, K. Xinidis, E. Markatos, and A. Keromytis. Detecting targeted attacks using shadow honeypots. In Proceedings of 14th USENIX Security Symposium, August 2005.
4. M. Bailey, E. Cooke, D.Watson, F. Jahanian, and N. Provos. A hybrid honeypot architecture for scalable network monitoring