BotNet Detection by Monitoring Network Traffic Using K-ANN Algorithm

Arvind Kishanrao Rathod

1Lecturer in Computer Engineering, Department of Computer Engineering, Government Polytechnic Jintur, District – Parbhani, Maharashtra, India

Abstract: A botnet is a collection of computers or other devices such as smartphones connected to the Internet whose security have been compromised or breached and are being controlled remotely by an intruder (the botmaster) via malwares (malicious software) called bots. The controller of botnet is able to direct the activities of these compromised computers through communication channels formed by standards. In order to secure network from those bot network and to provide secure network for organization. Due to increase of the cyber threat, I have made cyber security paramount for protecting personal and private information to get more secure network in the real time information; I have to provide the most highly secure network environment, network traffic monitoring and the most important threats detection system that is varied from the enterprise networks.

Keywords: SMB, DDoS, C&C, P2P

1. Introduction

As companies continue to send vital information on the internet that can affect the outcome of governments, markets, and industries alike, it’s more important than ever to have a solid security strategy in place. The marketplace has reflected this need over the last ten years, with a growing number of network security monitoring tools being developed that offer threat and intrusion detection. Companies like Dell introduced their SecureWorks software, and products like FireEye and Palo Alto also gained traction as new ways to protect your network from intrusion. Vendors like Cisco and Fortinet even went as far as building intrusion prevention software modules into their hardware to support the ever-increasing need for security.

Unfortunately, these are all enterprise-level security solutions that only customers with the right budget can afford. What about small-to-medium sized businesses (SMBs) that still need a way to identify potential security threats? Hackers are increasingly targeting smaller businesses, instinctively knowing that those companies will be the least equipped to handle attacks. What’s more, maintaining a secure network isn’t purely a business concern anymore. Hackers can also use a home network as a vehicle to access larger networks, stealing critical information from anyone at any time.

Hackers make it their job to continuously figure out different ways to access networks and steal data. In today’s world, we use web servers to enterimportant, sensitive information all the time. If these servers are not secured properly, a hacker can easily access them. We’ve recently seen a huge rise in cyber-attacks on various institutions in the United States. Because of this, companies that were once nonchalant about securing their infrastructure are now going to great lengths to increase their network security. Proactive monitoring of your network provides the details needed to fix performance problems in network devices, services, applications, connections, and traffic. Also, botnets have become one of the biggest threats to security systems today. Their growing popularity among cybercriminals comes from their ability to infiltrate almost any internet-connected devices. Botnets are also becoming a larger part of cultural discussions around cyber security. The use of botnets to mine cryptocurrencies like Bit coin is a growing business for cyber criminals. It’s predicted the trend will continue, resulting in more computers infected with mining software and more digital wallets stolen. Aside from being tools for influencing elections and mining crypto currencies, botnets are also dangerous to corporations and consumers because they’re used to deploy malware, initiate attacks on websites, steal personal information, and defraud advertisers. Its clear botnets are bad, but what are they exactly? And how can you protect your personal information and devices? Step one understands how bots work. Step two is taking preventative actions. To better understand how botnets function, consider that the name itself is a blending of the words “robot” and “network”. In a broad sense, that’s exactly what botnets are: a network of robots used to commit cybercrime. The cyber criminals controlling them are called botmasters or bot herders.

To build a botnet, botmasters need as many infected online devices or “bots” under their command as possible. The more bots connected, the bigger the botnet. The bigger the botnet, the bigger the impact. So size matters. The criminal’s ultimate goal is often financial gain, malware propagation, or just general disruption of the internet.

Cybercriminals use botnets to create a similar disruption on the internet. They command their infected bot army to overload a website to the point that it stops functioning and/or access is denied. Such an attack is called a denial of service or DDoS.

Botnets aren’t typically created to compromise just one individual computer; they’re designed to infect millions of devices. Bot herders often deploy botnets onto computers through a Trojan horse virus. The strategy typically requires users to infect their own systems by opening email attachments, clicking on malicious pop up ads, or downloading dangerous software from a website. After infecting devices, botnets are then free to access and modify.
personal information, attack other computers, and commit other crimes.

More complex botnets can even self-propagate finding and infecting devices automatically. Such autonomous bots carry out seek-and-infecct missions, constantly searching the web for vulnerable internet-connected devices lacking operating system updates or antivirus software.

Botnets are difficult to detect. They use only small amounts of computing power to avoid disrupting normal device functions and alerting the user. More advanced botnets are even designed to update their behavior so as to thwart detection by cybersecurity software. Users are unaware they’re connected device is being controlled by cyber criminals. What’s worse, botnet design continues to evolve, making newer versions harder to find.

Botnets take time to grow. Many will lay dormant within devices waiting for the botmaster to call them to action for a DDoS attack or for spam dissemination. Aside from DDoS attacks, botmasters also employ botnets for other malicious purposes. Cybercriminals can use the combined processing power of botnets to run fraudulent schemes. For example, botmasters build ad fraud schemes by commanding thousands of infected devices to visit fraudulent websites and “click” on ads placed there. For every click, the hacker then gets a percentage of the advertising fees.

1.1 Botnet Structures

Botnet structures usually take one of two forms, and each structure is designed to give the botmaster as much control as possible.

1.1.1 Client-server model

The client-server botnet structure is set up like a basic network with one main server controlling the transmission of information from each client. The botmaster uses special software to establish command and control (C&C) servers to relay instructions to each client device. While the client-server model works well for taking and maintaining control over the botnet, it has several downsides: it’s relatively easy for law enforcement official to location of the C&C server, and it has only one control point. Destroy the server, and the botnet is dead.

1.1.2 Peer-to-peer

Rather than relying on one centralized C&C server, newer botnets have evolved to use the more interconnected peer-to-peer (P2P) structure. In a P2P botnet, each infected device functions as a client and a server. Individual bots have a list of other infected devices and will seek them out to update and to transmit information between them. P2P botnet structures make it harder for law enforcement to locate any centralized source. The lack of a single C&C server also makes P2P botnets harder to disrupt. Like the mythological Hydra, cutting off the head won’t kill the beast. It has many others to keep it alive.

2. Related Work

For preventing botnet infection I have developed a comprehensive strategy; the proposed system consists of a botnet is a collection of computers which is connected to the Internet they have been controlled remotely by an intruder (the bot-master) via malicious software called bots. The system consists of different attacks DDOS, HTTP attack, IP spoofing, HTTP header, HTTP header with multiple IP. Middleware consists of botnet detection system which consists of java packet cap-true library and Win P cap library, which is meant to detect network traffic through these libraries. While a significant amount of research has been accomplished on botnet analysis and detection, several challenges remain unaddressed, such as the

Ability to design detectors which can cope with new forms of botnets. Detect botnet based on traffic behavior analyzing network traffic behavior using machine learning. Traffic behavior analysis methods do not depend on the payload of the packet, which means that they can work with encrypted network communication protocols. Network traffic information can retrieved from network devices without affecting network performance. The final stage of our architecture is to detect bot across net-work by applying K-ANN algorithm on the real-time packet information which is gathered by JPCAP and WinPcap library.

![Figure: Architecture](image)

2.1 Creating TCP Packet Dumper

This project makes use of JPCAP library to capture packets and analyze and filter by its type. The application consists of following modules. Using JPCap library dumps the all TCP requests (packet) into the dump file. Dump file size is maximum 2048KB. Dump file created on the basis following attribute: packet Index, Timeval, source Address, source Hardware Address, source Port, destination Address, destination Hardware Address, destination Port, sequence Number, acknowledgement-Number, flags Present, packet Priority, packet Length, offset, Time To Live.

2.2 Apply TCP Packet Filtering.

Then we can apply the TCP packet filter on the dump files only for the incoming TCP request.
2.3 Web Application Firewall.

This project makes use of JPCap library to capture packets and analyze and filter by its type. The application consists of following modules.

2.3.1 TCP Packet Analyzer

It performs the analysis on dump files and gives the following attribute to KANN (Kohonen Artificial Neural Network).
- Hardware Address.
- Count of Packet from the Different IP.
- Average Length of TCP packet.
- Average difference in consecutive requests (delta time).
- Request Ratio Number of self-packets/total packets.

2.3.2 Apply K-ANN

(Kohonen Artificial Neural Network) TCP Packet analyzer output use as the Kohonen Artificial Neural Network input the data set is trained on the TCP Packet Analyzer attribute.

2.3.3 Deployment Diagram

This diagram shows us how the system is deployed in the real world. Data collected from different network devices. WinPCap and JpCap Libraries are used to extract the information. This extracted information is analyzed from K-ANN algorithm.

2.4 Kohonen Artificial Neural Network

Step 1: Then-dimensional weight vectors w1,w2, . . . ,wm of the m computing units are selected at random. An initial radius r, a learning constant r, and a neighborhood function are selected.

Step 2: Select an input vector x using the desired probability distribution over the input space.

Step 3: The unit k with the maximum excitation is selected (that is, for which the distance between wi and x is minimal, i = 1, . . . ,m).

Step 4: The weight vectors are updated using the neighborhood function and the update

Rule
Wi ←Wi + (i, k)(x - Wi ), For i = 1, . . . , m.

Step 5: Stop if the maximum number of iterations has been reached; otherwise modify (i) and ( ) as scheduled and continue with step 1.

2.5 Snapshots

2.5.1 Admin Dash: This is the Dashboard of Network Analyzer which displays main home screen of Net Analyzer web application it is having a tab which directs pages like Traffic Viewer, Statistics and returns to the main screen.

2.5.2 Traffic Viewer

This dashboard shows the real time traffic from the network in the index format it creates traffic log which is also called as the dump file. It shows traffic log in different categories like Destination IP, Source IP, Flags, Length, Offset, and TTL.

2.5.3 File Analysis

This dashboard shows the real-time traffic threat by monitoring the Derived packet attribute from the network with the help back-propagation and error calculation method. It shows the destination port or IP address of the bot computer to detect the source of the attack. This condition is also called as HTTP attack. It also shows traffic log in
different categories like Destination IP, Source IP, Flags, Length, Off-set, TTL.

2.5.4 Multiple IP: This dashboard shows the multiple IP http attacks where the attacker try to flood your a system with multiple IP which will assume you that the request is coming from different ip or system but it doesn’t. Multiple ip column shows the result as “true”.

3. Conclusion
I have shown through different sets of experiments that our proposed model addresses adequately two of the above challenges, namely, early detection and novelty detection. my proposed model allows detecting bot activity in both the command and control and attack phases based on the observation of its network flow characteristics for specific time intervals. I emphasize the detection of the command and control phase because we would like to detect the presence of a bot early before any malicious activities can be performed, and we use the concept of time intervals to limit the duration we would have to observe any particular flow before I may raise our suspicions about the nature of the traffic.

References

Author Profile
Arvind Kishanrao Rathod, completed B.E in Computer Science and Engineering at MIT College of Engineering, Dr. B.A.M. University, Aurangabad. Presently Working as a Lecturer in Computer Engineering at Government Polytechnic, Jintur District Parbhani, Maharashtra, India, Has 16 Years of Teaching Experience.