AN IMPLEMENTATION OF INTRUSION DETECTION AND PREVENTION SYSTEMS

Dr. G.N.K.Suresh Babu[1], Dr. M. Kumarasamy[2]
[1]Professor, Department of Computer Science, Acharya Institute of Technology, Bangalore
[2]Professor, Department of Informatics, Wollega University, Ethiopia /Africa

ABSTRACT

The purpose of the paper is to develop a two-stage approach for the important cyber-security problem of detecting the presence of a botnet and identifying the compromised nodes (the bots), ideally before the botnet becomes active. The expansion of mobile computing devices such as laptops and Personal digital assistants (PDAs) increased the demand for the Wireless Local Area Networks (WLANs) in the corporate world. In last decade, there was a huge increase in the deployment of the wireless LANs in the corporate and this growth is drastically increases every year due to its low cost, strong performance and ease of deployment. Although many enterprises are highly concerned about the security issues such as unauthorized use of service and privacy in their Wireless LANs, they are unaware or unsure about providing a strong security to their wireless network. Within the past 10 years, wireless LANs become more prevalent at home, most of the computer users are getting adapted to laptops where they setup wireless networks to connect internet. Unlike wired LANs, wireless LANs is more vulnerable to intrusions, misuse and attacks due to the fact that the data is transmitted across the air in clear text which is easily accessible. Wireless hacking is one of the major issues in the wireless networking where there are numerous and potentially dangerous threats that includes interception, unauthorized monitoring of wireless traffic, client to client attacks, jamming, session hijacking, etc.

Keywords: Intruders, Attacks, Virus, Wireless, Hackers

1. INTRODUCTION

Intrusion detection and prevention mechanisms can detect malicious activities performed by external or internal attackers, by monitoring and analyzing network activities. The existing IDS architectures for ad hoc networks can be classified into stand-alone, cooperative and hierarchical. In stand-alone architecture, every node in the network performs detection based on its local data using an IDS agent installed on it. In cooperative architecture, each node has an IDS agent that communicates and collaborates with other nodes' agents, forming a global intrusion detection to resolve inconclusive detections. Hierarchical IDS is a sm

ort of cooperative architecture suited to multi-layered networks. In this architecture, the network is divided into clusters, where some nodes are selected as cluster heads to undertake more responsibility than other cluster members. Each cluster member performs local detection, while cluster heads perform global detection (Sen, 2010). This paper focuses on both home and enterprise wireless local area networks. It starts with a research on the various threats, attacks and vulnerabilities in the WLANs. Multi-hop ad hoc networks are often secured by using either cryptography or intrusion detection systems (IDSs). Cryptographic approaches can protect ad hoc networks against external attackers using node authentication and data encryption. However, these techniques cannot prevent insider attacks, consumes considerable

resources and have their associated problems such as key issuing and management.

2. INTRUSION DETECTION AND PREVENTION SYSTEM

IDS/IPS devices need two things to provide an effective layer of security. The devices must be tuned to the network they monitor and tuned-in to the latest threats. Getting the maximum ROI from your investment in IDS/IPS is easier with a bit of expert help. The tough part of big data is analyzing it to know what action to take. IDS devices can generate thousands of alerts daily, many of which are false positives, which can send your team off to chase ghosts. Keeping your IDS/IPS devices tuned, up-to-date and monitored appropriately given new emerging threats can become a heavy burden for limited security resources. Shifting this burden to a managed service staffed with security device experts can offer needed relief, along with improved insights that help you take the right action to remediate identified threats. Fig 1 represents how firewall can be used for intrusion detection.

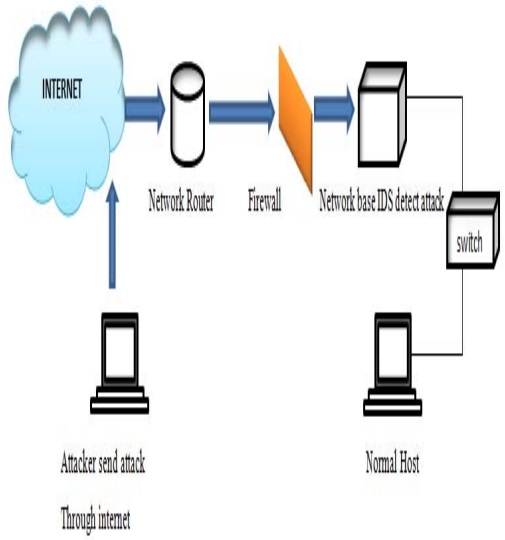

Fig. 1 Intrusion Detection Using Firewall Systems

3. INTRUSION DETECTION TOOLS

There are a wide variety of IDSs available, ranging from antivirus to hierarchical systems, which monitor network traffic. The most common ones are listed below.

- ***NIDS:*** Network intrusion detection systems are placed at highly strategic points within the network to monitor inbound and outbound traffic from all devices in the network. But scanning all traffic could lead to the creation of bottlenecks, which impacts the overall speed of the network.
- ***HIDS:*** Host intrusion detection systems run on separate machines or devices in the network, and provide safeguards to the overall network against threats coming from the outside world.
- ***Signature based IDS:*** Signature based IDS systems monitor all the packets in the network and compare them against the database of signatures, which are pre-configured and pre-determined attack patterns. They work similar to antivirus software.
- ***Anomaly based IDS:*** This IDS monitors network traffic and compares it against an established baseline. The baseline determines what is considered normal for the network in terms of bandwidth, protocols, ports and other devices, and the IDS alerts the administrator against all sorts of unusual activity.
- ***Passive IDS:*** This IDS system does the simple job of detection and alerting. It just alerts the administrator for any kind of threat and blocks the concerned activity as a preventive measure.
- ***Reactive IDS:*** This detects malicious activity, alerts the administrator of the threats and also responds to those threats.

4. OPEN SOURCE NETWORK INTRUSION DETECTION TOOLS

4.1Snort

Snort is a free and open source network intrusion detection and prevention tool. It was created by Martin Roesch in 1998. The main advantage of using Snort is its capability to perform real-time traffic analysis and packet logging on networks. With the functionality of protocol analysis, content searching and various pre-processors, Snort is widely accepted as a tool for detecting varied worms, exploits, port scanning and other malicious threats. It can be configured in three main modes — sniffer, packet logger and network intrusion detection. In sniffer mode, the program will just read packets and display the information on the console. In packet logger mode, the packets will be logged on the disk. In intrusion detection mode, the program will monitor real-time traffic and compare it with the rules defined by the

user.
Snort can detect varied attacks like a buffer overflow, stealth port scans, CGI attacks, SMB probes, OS fingerprinting attempts, etc. It is supported on a number of hardware platforms and operating systems like Linux, OpenBSD, FreeBSD, Solaris, HP-UX, MacOS, Windows, etc.

4.2SecurityOnion
Security Onion is a Linux distribution for intrusion detection, network security monitoring and log management. The open source distribution is based on Ubuntu and comprises lots of IDS tools like Snort, Suricata, Bro, Sguil, Squert, Snorby, ELSA, Xplico, NetworkMiner, and many others. Security Onion provides high visibility and context to network traffic, alerts and suspicious activities. But it requires proper management by the systems administrator to review alerts, monitor network activity and to regularly update the IDS based detection rules. Security Onion has three core functions:

- Full packet capture
- Network based and host based intrusion detection systems
- Powerful analysis tools

Full packet capture: This is done using netsnifff-ng, which captures all network traffic that Security Onion can see, and stores as much as your storage solution can hold. It is like a real-time camera for networks, and provides all the evidence of the threats and malicious activities happening over the network.

Network-based and host-based IDS: It analyses the network or host systems, and provides log and alert data for detected events and activity. Security Onion has varied IDS options like rule-driven IDS, analysis-driven IDS, HIDS, etc.

Analysis tools: In addition to network data capture, Security Onion comprises various tools like Sguil, Squert, ELSA, etc, for assisting administrators in analysis.
Security Onion also provides diverse ways for the live deployment of regular standalone, server-sensor and hybrid monitoring tools.

4.3OpenWIPS-NG
OpenWIPS-NG is a free wireless intrusion detection and prevention system that relies on sensors, servers and interfaces. It basically runs on commodity hardware. It was developed by Thomas d'Otrepe de Bouvette, the creator of Aircrack software. OpenWIPS uses many functions and services built into Aircrack-NG for scanning, detection and intrusion prevention. The three main parts of OpenWIPS-NG are listed below:

Sensor: Acts as a device for capturing wireless traffic and sending the data back to the server for further analysis. The sensor also plays an important role in responding to all sorts of network attacks.
Server: Performs the role of aggregation of data from all sensors, analyses the data and responds to attacks. Additionally, it logs any type of attack and alerts the administrator.
Interface: The GUI manages the server and displays the information regarding all sorts of threats against the network.

4.4Suricata
Suricata is an open source, fast and highly robust network intrusion detection system developed by the Open Information Security Foundation. The Suricata engine is capable of real-time intrusion detection, inline intrusion prevention and network security monitoring. Suricata consists of a few modules like Capturing, Collection, Decoding, Detection and Output. It captures traffic passing in one flow before decoding, which is highly optimal. But unlike Snort, it configures separate flows after capturing and specifying how the flow will separate between processors.

4.5BroIDS
BroIDS is a passive, open source network traffic analyser developed by Vern Paxson, and is used for collecting network measurements, conducting forensic investigations, traffic base lining and much more. BroIDS comprises a set of log files to record network activities like HTTP sessions with URIs, key headers, MIME types, server responses, DNS requests, SSL certificates, SMTP sessions, etc. In addition, it provides sophisticated functionality for the analysis and detection of threats, extracting files from HTTP sessions, sophisticated malware

detection, software vulnerabilities, SSH brute force attacks and validating SSL certificate chains. BroIDS is divided into the following two layers.

Bro Event Engine: This does the task of analysing live or recorded network traffic packs using C++ to generate events when something unusual happens on the network.
Bro Policy Scripts: These analyse events to create policies for action, and events are handled using policy scripts such as sending emails, raising alerts, executing system commands and even calling emergency numbers.

4.6OSSEC

OSSEC is a free and open source host based IDS that performs varied tasks like log analysis, integrity checking, Windows registry monitoring, rootkit detection, time-based alerting and active response. The OSSEC system is equipped with a centralised and cross-platform architecture allowing multiple systems to be accurately monitored by administrators. The OSSEC system comprises the following three main components.

- **Main application:** This is a prime requirement for installations; OSSEC is supported by Linux, Windows, Solaris and Mac environments.
- **Windows agent:** This is only required when OSSEC is to be installed on Windows based computers/clients as well as servers.
- **Web interface:** Web based GUI application for defining rules and network monitoring.

4.7 Open Source Tripwire

Open Source Tripwire is a host based intrusion detection system focusing on detecting changes in file system objects. On the first initialisation, Tripwire scans the file system as instructed by the systems administrator and stores the information of each file in a database. When files are changed and on future scans, the results are compared with the stored values and changes are reported to users. Tripwire makes use of cryptographic hashes to detect changes in files. In addition to scanning file changes, it is used for integrity assurance, change management and policy compliance.

4.8 AIDE

AIDE (Advanced Intrusion Detection Environment) was developed by Rami Lehti and Pablo Virolainen. It is regarded as one of the most powerful tools for monitoring changes to UNIX or Linux systems. AIDE creates a database via regular expression rules that it finds from the *config* files. On initialising the database, it is used to verify the integrity of files. Some of the most powerful features of AIDE are as follows:

- Supports all kinds of message digest algorithms like MD5, SHA1, RMD160, TIGER, SHA256 and SHA512.
- Supports POSIX ACL, SELinux, XAttra and Extended File System.
- Powerful regular expression support to include or exclude files and directories for monitoring.
- Supports various operating system platforms like Linux, Solaris, Mac OS X, UNIX, BSD, HP-UX, etc.

5 PROPOSED METHODOLOGY

We begin the implementation with traffic filtering, which eliminates all Non-P2P Clients. Followed by Fine grained P2P clients where those Non-P2P clients that escaped the traffic filter stage are eliminated here. This is followed by Coarse grained detection of P2P Bots where most of the legitimate P2P clients are identifies and eliminated. Finally the few legitimate P2P clients present are identified and eliminated in Fine grained P2P Bot detection. Those hosts that are left after the filtering process are the bots present in the system.

Module description

We have four modules in this project and they are:

1. Traffic Filtering
a) Network Monitoring b) DNS Traffic

2. Fine Grained Detection of P2P Clients
a) Cluster Formation based on protocol

3. Coarse Grained Detection of P2P Bots
a) Calculate Start and End Time Stamp

b) Calculate Active time
c) Compare Active time with System time

4. Fine Grained Detection of P2P Bots

a) EuclideanDistance Calculation
b) Peer Evaluation

Algorithm and detailed explanation for each module

1. Traffic Filtering

This module will accept the input from the network traffic. The input will be a set of hosts present in the network and these hosts include P2P clients, P2P Bots and Non-P2P traffic. The Traffic Filter component aims at filtering out network traffic that is unlikely to be related to P2P communications. This is accomplished by passively analyzing DNS traffic, and identifying network flows whose destination IP addresses were previously resolved in DNS responses and then eliminating it from the system.
Input: Network traffic

Output: DNS Traffic Eliminated from the initial input traffic

2. Fine Grained Detection of P2P Clients

This component is responsible for detecting P2P clients by analyzing the remaining network flows after the Traffic Filter component. Each of the host is classified into clusters based on the network flow (TCP or UDP). Once all the hosts are grouped into clusters, the Euclidean distance between the two flows are calculated for each host.The average distance for each of the cluster is calculated and summarized. If the average distance does not exceed a certain threshold value, then the corresponding host is discarded.
Input: Network traffic that contain only those hosts with a legitimate IP address

Output: P2P clients (including Bots and legitimate clients) are retained

3. **Coarse Grained Detection of P2P Bots**

The output from the previous model will have only those clients that are P2P. This component is responsible for identifying the bots. The active time of each client is calculated. Also, the system time of the underlying host is calculated. By comparing the active time and the system time, the P2P clients can be distinguished from the bots. The feature that is leveraged is that, a client that has its active time comparable with the system time is a bot as a bot must be active on the system for a long time to listen to the system communication. Those clients that have active time not comparable with the system are identified as legitimate P2P clients and are discarded.
Input: P2P clients (including Bots and legitimate clients)

Output: Legitimate P2P clients

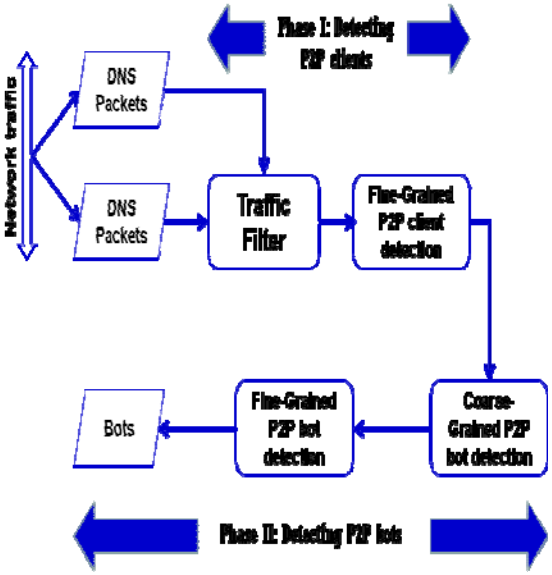

Fig.2 Proposed Methodology

6.CONCLUSION

Before evaluating IDPS products, organizations should first define the general requirements that the Products should meet. The features provided by IDPS products and the methodologies that they use vary considerably, so a product that best meets one organization's requirements might not be suitable for meeting another organization's requirements. Information security has become a legitimate concern for both organisations and computer users due to the growing confidence on computers and electronic transactions. Different techniques are used to support the security of an organization against threats or attacks. On the other side, attackers are discovering new techniques and ways to break these security policies. Firewall, antivirus and antispyware are limited to provide security to the system against threats. This paper concludes that both the intrusion detection systems and preventions systems still need to be improved to ensure an unfailing security for a network. They are not reliable enough and they are difficult to administer. It is

obvious that these systems are now essential for companies to ensure their security. To assure an effective computerized security, it is strongly recommended to combine several types of detection system. Our proposed methodology combines some of the best features of existing IDS tools and we try to give some solutions to the prevention systems.

REFERENCES

[1] U. A. Sandhu, S. Haider, S. Naseer, O. U. Ateeb, “A Survey of Intrusion Detection & Prevention Techniques”, International Conference on Information Communication and Management IPCSIT: IACSIT Press, Singapore 2011.

[2] K. Scarfone and P. Mell, Guide to Intrusion Detection and Prevention Systems (IDPS), Recommendations of the National Institute of Standards and Technology: NIST Special Publication, February 2007.

[3] B. Menezes, Network Security and Cryptography: CENGAGE Learning, Chapter 14, 18, 19, 21, 22, 24, 2010.

[4] J. R. Vacca, Computer and information security handbook: Morgan Kaufmann Series in Computer Security, First edition, May 4, 2009.

[5] Langin, C. L. A SOM+ Diagnostic System for Network Intrusion Detection. Ph.D. Dissertation, Southern Illinois University Carbondale (2011).

[6] F. Cikala, R. Lataix, S. Marmeche", The IDS/IPS. Intrusion Detection/Prevention Systems ", Presentation, 2005.

[7] Hervé Debar and Jouni Viinikka, "Intrusion Detection,: Introduction to Intrusion Detection Security and Information Management", Foundations of Security Analysis and Design III, Reading Notes in to Compute Science, Volume 3655, 2005. pp. 207-236.

[8] Abdullah A. Mohamed, “Design Intrusion Detection System Based On Image Block Matching”, International Journal of Computer and Communication Engineering, IACSIT Press, Vol. 2, No. 5, September 2013.

[9] Denning, Dorothy E., "An Intrusion Detection Model," Proceedings of the Seventh IEEE Symposium on Security and Privacy, May 1986, pages 119–131

[10] Karen Scarfone & Peter Mell, “Guide to Intrusion Detection and Prevention Systems (IDPS)”, NIST Special Publication on Computer security, February 2007.

[11 Lunt, Teresa F., "Detecting Intruders in Computer Systems," 1993 Conference on Auditing and Computer Technology, SRI International.