Scientific Journal of Impact Factor (SJIF): 5.71

e-ISSN (O): 2348-4470
p-ISSN (P): 2348-6406

International Journal of Advance Engineering and Research Development

Volume 5, Issue 02, February -2018

DETECTING THREATS USING PORT SCANNING

Dr.P.Rajkumar[1], P.Aswathi[2], K.Mohana Prabha[3], S.Santhya[4]

[1]*Associate Professor , Computer Science and Engineering, Info Institute of Engineering, TamilNadu*
[2,3,4]*UG Scholar, Computer Science and Engineering, Info Institute of Engineering, TamilNadu*

Abstract- *Port scanner is a piece of software designed to search a network host for open ports. This is often used by administrators to check the security of their networks and by crackers to compromise it. To port scan a host is to scan for listening ports on a single target host. To port sweep is to scan multiple hosts for a specific listening ports. Some port scanner only scan the most common or most commonly vulnerable port numbers on a given host. Port scanning can however also be used by those who intend to compromise security. In proposed system, an alternative engine is implemented to automatically block specialized tool scans, namely PSAD. To carry out this work, a virtual network environment is configured as an experimenting platform with port scan attacks. To neutralize such attacks, security mechanism is performed that takes the data reported by the PSAD and using parameterized variables automatic locks become viable including custom record and notifications. The result of alternative engine is faster and more reliable than the tools previously used.*

Keywords- *Port Scanner, Threats, PSAD, Vulnerability, Host, Ports*

I. INTRODUCTION

Port scanning is one of the most popular techniques attackers use to discover services that they can exploit to break into systems. All systems that are connected to a LAN or the internet via a modem run services that listen to well-known and not so well-known ports. By port scanning, the attacker can find the following information about the targeted systems what services are running, what users own those services, whether anonymous logins are supported, and whether certain network services require authentication. Port scanning is accomplished by sending a message to each port, one at a time. The kind of response received indicates whether the port is used and can be probed for further weaknesses.

Port scanners are important to network security technicians because they can reveal possible security vulnerabilities on the targeted systems. Just as port scans can be ran against your systems, port scan can be detected and the amount of information about open services can be limited utilizing the proper tools. Every publically available systems has ports that are open and available for use. The object is to limit the exposure of open ports to authorized users and to deny access to the closed ports.

The threat detection feature consists of different levels of statistics gathering for various threats, as well as scanning threat detection, which determines when a host is performing a scan.

User can optionally shut any hosts determined to be a scanning threat. Threat detection statistics can help you manage threats to your ASA, for example, if you enable scanning threat detection, then viewing statistics can help to analyze the threat.
User can configure two types of threat detection statistics. Basic threat detection statistics that include information about attack activity for the system as a whole. Basic threat detection statistics are enabled by default and have no performance impact.

Advanced threat detection statistics track activities at an object level, so the ASA can report activity for individual hosts, ports, protocols or access list. Advanced threat detection statistics can have a major performance impact, depending on the statistics gathered, so only the access list statistics are enabled by default. A vulnerability scanner is software application that access security vulnerabilities in network or host systems and produces a set of scan results. However, because both administrators and attackers can use the same tool for fixing or exploiting a system, administrators need to conduct a scan and fix problems before the attacker can do the same scan and exploit any vulnerability found.

This project provide a general overview of vulnerability scanners. In addition, many vulnerability scanners rely on “plug-ins” to determine potential vulnerabilities. Plug-ins are part of knowledge database (or scan database) of the

vulnerabilities that the scanner is capable of detecting. These databases may be named differently (such as "Scanning Profile") in different scanner products.

A scanner can only check for those vulnerabilities that is "Knowns", by cross checking with the presence of its corresponding installed plug-in set. It cannot identify those vulnerabilities that don't have a plug-in. Not all scanners need plug-ins as they just scan a target range ports.

II. EXISTING PORT SCAN SYSTEM

Attackers employ various technologies to launch attacks, such as Denial-of-service (DOS), BotNet , worm and virus etc. The first step of these attacks is to discover vulnerable victim hosts. Nearly all attackers perform port scanning to find vulnerabilities on victim hosts. Most of existing fast-replicated viruses and worms perform port scanning to discover and infects targets. Hence, it is crucial to study port scanning and explore whether the last advances in technologies have changed the horizon of port scanning including how to perform port scanning, expedite scanning speed, conduct port scanning from multiple and defend against modern port scanners.

Different network protocols employ different ports. Vulnerabilities exist in all protocols. Hence, to gather information completely, port scanners have to perform scanning for a large number of ports. The size of the port space is 65535 ports 0 to 1023 are well- knows ports, ports 1024 to 49151 are registered ports, and ports 49152 to 65535 are dynamic or private ports. Ports scanners must run extremely fast. Port scanners have employed sophisticated techniques to expedite port scanning. For example, worms can search vulnerabilities on a commonly used port (e.g. port 21 used by FTP, and port 443 used by HTTPS).

However, a typical complete port scan is time-consuming. For example, a 65536 port UDP scan for one target host could take more than 18 hours. Attackers typically perform port scanning independently, without coordination, to find victim hosts.

If port scanning software packages are run on multiple machines without coordination, their search spaces will overlap significantly. The overlap causes reduction in the performance of the scanning. The network connections used by the port scanners could get congested. The buffer size of the network software may not be large enough to hold all the incoming data. The processing speed of the computer may not be enough to analyze responses from all the network. After all scanning activities end, all the computers involved in the scanning must communicate to each other and finalize the search results. Problems arise when their results differ. Such differences are hard to analyze, due to the fast changing nature of computer networks. As defense technologies evolve, port scanners that exhibit unusual network behaviors, such as sending requests to all IP addresses in class B network are more likely to be detected. Such detection will likely disable the machine performing the scanning immediately and trigger chained detections of all other machines involved. Given the fact that virtually all networks are protected by firewalls, filters and monitors, a simple deployment of identical port scanning software packages to all computers involved in the scanning is not acceptable.

2.1 DRAWBACKS

- Pre-existing system can scan only for the open ports and not detect the vulnerabilities
- If any defect occurs then there is intent to find for threats, where there is additional requirement for software
- Detected threats were not resolved

III. PROPOSED SYSTEM

The Internet today is a complex entity comprised of diverse network users, and resources. Most of the users are oblivious to the design of the Internet and its components and only use the services provided by their operating system or applications. However there is a small minority of advanced users who use their knowledge to explore potential system vulnerabilities. Hackers can compromise the vulnerable hosts and can either take over their resources or use them as tools for future attacks. With so many different protocols and countless implementations of each for different platforms, the launch of an effective attack often begins with a separate process of identifying potential victims.
One of the popular methods for finding susceptible hosts is port scanning can be defined as "hostile internet searches for open 'doors', or ports, through which intruders gain access to computers". This technique consist of sending a message to a port and listening for an answer. The received response indicates the port status and can be helpful in determining a host's operating system and other information relevant to launching a future attack.
The goal of this project is to analyze and characterize port scanning traffic by defining a set of heuristics and applying them to the network trace data, we were able to isolate suspicious packets and group them into sets of scans and these sets were further analyzed to extract properties of the port scanning traffic and to collect relevant statistics.
The protocol stack that is most common on the Internet today is TCP/IP. In this system, hosts and host services are referenced using two components: an address and a port number. There are 65536 distinct and usable port numbers. Most

services use a limited range of numbers; these numbers will eventually become assigned by the IANA when the service becomes important enough. Some port scanners only scan the most common or most commonly vulnerable, port numbers on a given host.

3.1 BLOCK DIAGRAM OF PROPOSED SYSTEM

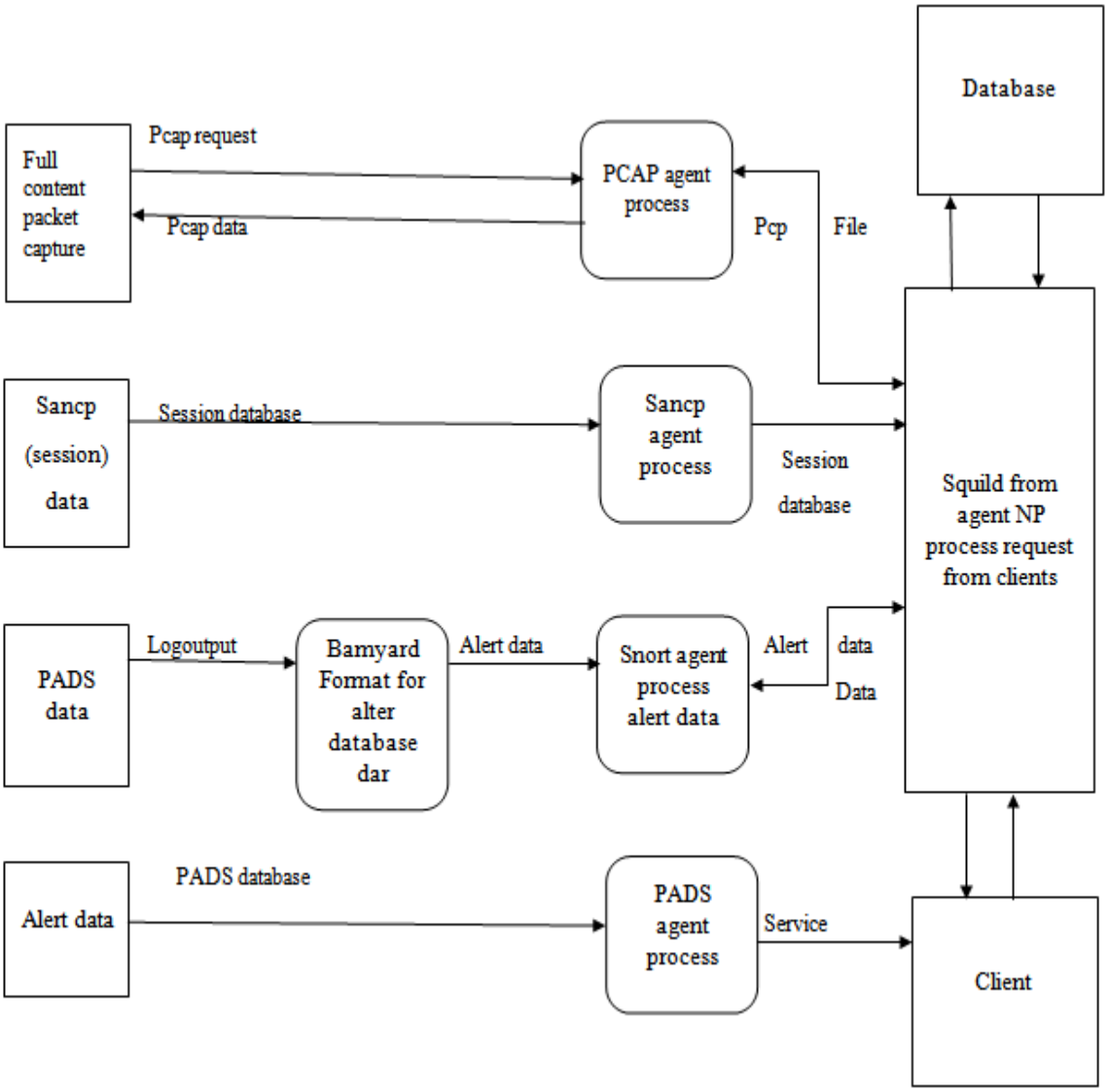

Figure 1. Architecture of proposed system

IV. COMPARISON BETWEEN EXISTING AND PROPOSED PORT SCANNING SYSTEM

4.1 PORT SCANS AND THEIR DETECTION METHODOLOGIES

Scanning of ports on a computer occurs frequently on the Internet. An attacker performs port scans of IP addresses to find vulnerable hosts to compromise. However, it is also useful for system administrators and other network defenders to detect port scans as possible preliminaries to more serious attacks. It is very difficult task to recognize instances of malicious port scanning. In general, a port scan may be an instance of a scan by attackers or an instance of a scan by network defenders. In this survey, we present research and development trends in this area.

Our project includes a discussion of common port scan attacks. We provide a comparison of port scan methods based on type, mode of detection mechanism used for detection, and other characteristics. This survey also reports on the available datasets and evaluation criteria for port scan detection approaches.

Port scanning is designed to probe a network host for open ports and other services available. It is useful for system administrators and other network defenders to detect port scans as useful technique for recognizing Precursors to serious attacks.

From the attacker's view point, a port scan is useful for gathering relevant information for launching a successful attack. It is of considerable interest to attackers to determine whether or not the defenders of a network are scanning ports regularly. Defenders do not usually hide their identity during port scanning while attackers do.

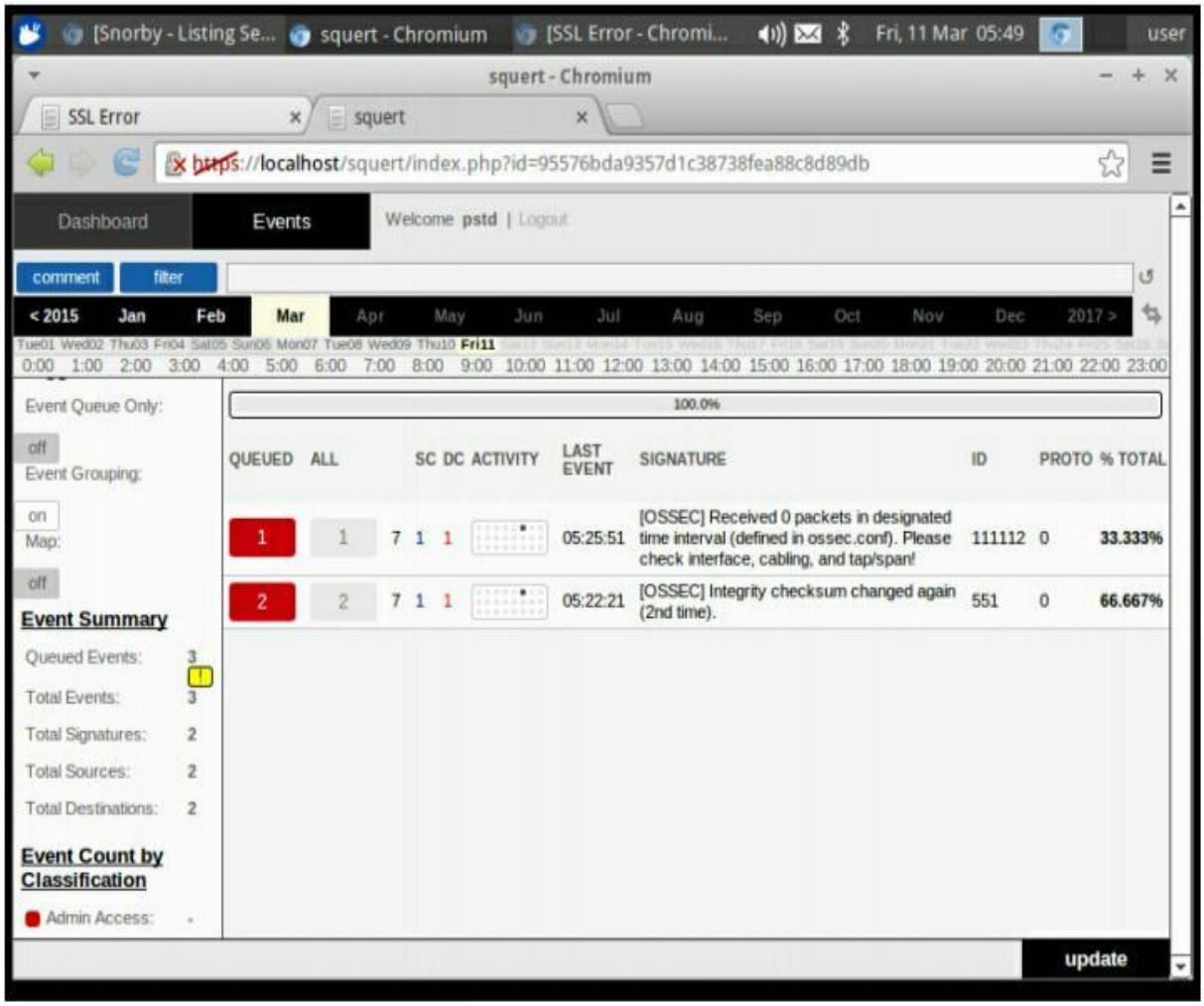

Figure 2. Port scan and Detection

4.2 COLLABORATIVE PORT SCANNING ATTACKS

Most network attackers perform port scanning individually, without synchronization, to find victim hosts. Such port scanning schemes suffer from two problems: first, there are too many duplicate scanning's and too much contention among different port scanners; second, a complete port scanning takes a long time to finish. In this project, we present a fast DHT-based collaborative port scanning scheme that aims to eliminate duplicate scanning minimize contention, and significantly increase the scanning speed.

In collaborative attacks, attackers communicate and collaborate with each other to launch much more powerful attacks. In the DHT-based collaborative port scanning scheme, attackers collaborate to search the network for ports that could be exposed to attacks. We propose different collaborative scanning strategies and analyze their advantages and disadvantages. WE discuss the static, dynamic, and hybrid target selection and allocation schemes.

We present the algorithms details and discuss the stop and revisit policy for the collaborative port scanners. We conduct Experiments to evaluate the performance and overhead of the collaborative port scanning strategies.

Experimental results suggests that the proposed collaborative port scanning system significantly increases the efficiency of port scanning and provide insights into many design and implementation issue.

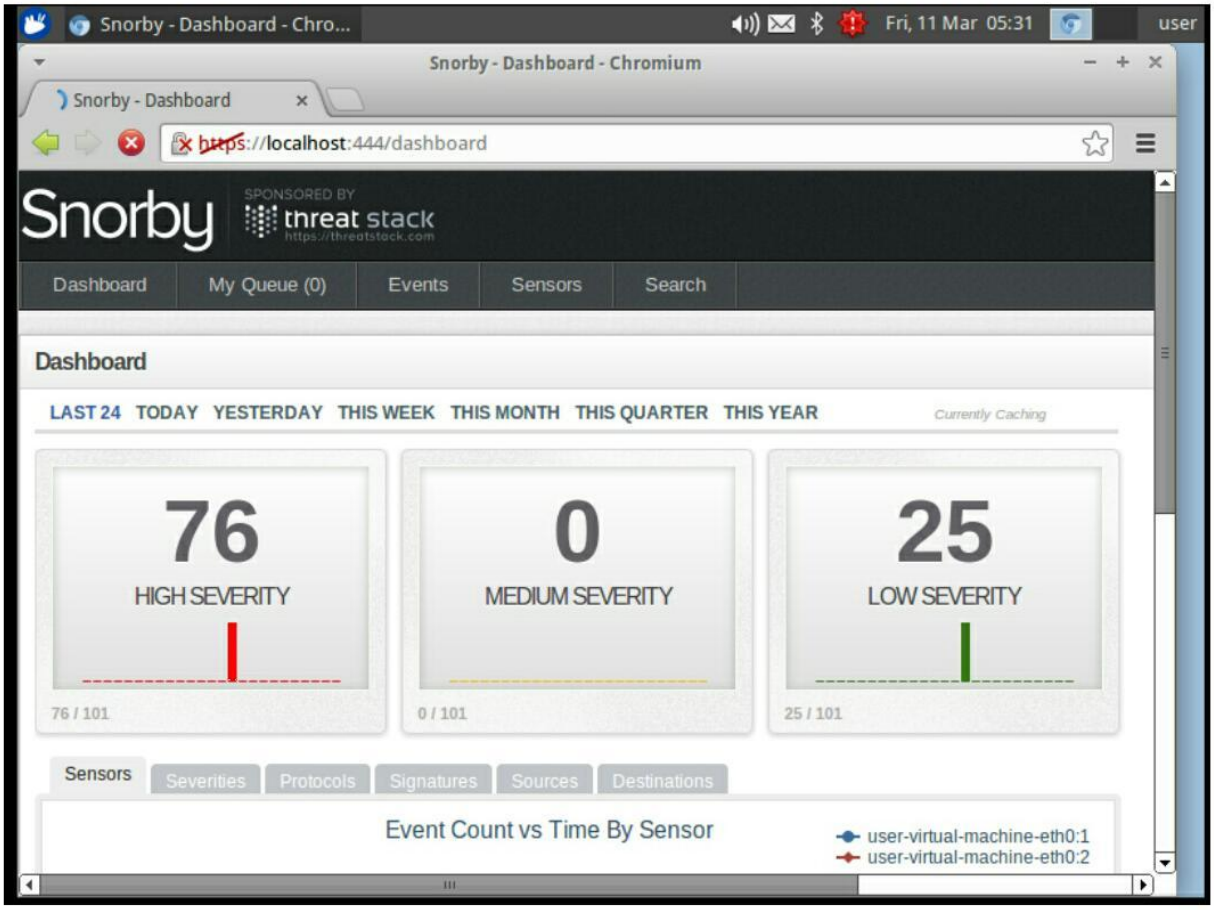

Figure 3. Severity of attacks

4.3 DETECT AND BLOCK PORT SCAN ATTACKS

IP networks are constantly harmed by several attack techniques such as port scans, denial of service, brute force attacks, etc., which can collapse the continuity of business services. To address this problem, this paper focuses on an alternative solution for detection, block, and prevention of port scanning attacks.

Particularly, this implementation is an alternate engine to automatically block specialized tool scans namely PSAD (PORT SCAN ATTACK DETECTOR), but it is conceptualized differently from the features that the program offers. To carry out this work we have designed and implemented a virtual network environment that is to be configured as an experimenting platform with port scan attacks. To neutralize such attacks, we performed a security mechanism that takes the data reported by the PSAD and using parameterized variables (block time and level of category) automatic locks become viable, including custom records and notifications via e-mail. To validate our solution, several tests of port scan attacks have been run on public and private networks. Then we have compared the performance our alternative engine with Clear OS (specialized security tool for Linux) and the PSAD. The results show that our alternative engine is faster and more reliable than the tools previously mentioned.

V. MODULE DESCRIPTION

5.1 PACKET CAPTURING

Download Wireshark for windows or Mac OS X. If you are using Linux or another UNIX-like system, you'll probably find Wireshark in its package repositories. For example, if you are using Ubuntu, you'll find Wireshark in the Ubuntu software centre . Just a quick warning: Many organizations don't allow Wireshark and similar tools on their networks. Don't use this tool at work unless you have permission.

After downloading and installing Wireshark, you can launch it and click the name of an interface under interface. List to start capturing packets on that interface. For example, if you want to capture traffic on the wireless network, click your wireless interface. You can configure advanced features by clicking Capture Options, but this isn't necessary for now.

5.2 ANALYZING THE THREATS

As the number of security threats to networks and servers grows security managers have turned to vulnerability analysis tools to identify a wide variety of potential problems on their networks. While host-oriented patch tools such as Update EXPERT from St. Bernard Software and HFNetChkPro from Shavlik Technologies focus on the myriad patches needed to keep Windows server up to date, network vulnerability analyzers look for more than just missing patches.

The tools can search for misconfigured application servers, such as Web servers and network components such as switches and routers ,that are vulnerable to known problems. They look for out-of-date applications especially those with known problems. And they often search for applications they are enabled by default but perhaps shouldn't be, such as RPC services on Unix or the UDP ECHO program on Windows NT/2000. Vulnerability analyzers are also security oriented, so they often look for "information leakage" from systems through DNS and other avenues, including SNMP and Windows registry.

Most vulnerability analyzers take a three-phase approach to testing:

- Given a network range by the security managers, the VA attempts to determine which IP address are in use. This phase usually includes tool such as ping.
- The VA attempts to determine which applications and services are running on these systems, and their configurations. The VA uses a variety of techniques, ranging from simply trying to connect(a port scan) to gathering and socket information out of SNMP
- The tool employs a long series of tests to find out if each system is susceptible to a particular known bug or problem. Smarter products iterate between phases two and three, learning more and using that information to launch additional tests. Others have ways of pruning their decision tree to save time and minimize the risk of overloading the target systems.

There are many variations with these three phases. Some products try to brute-force guess passwords on accounts. Others assume a friendly environment and connect to servers with administrative access to look for problems at the system level. Some are more devious, and will try to evade a network IDS.

5.3 VISUALIZATION

These approaches are used for virtualizing network traffic to detect whether the flow of network packet is an attack or normal behavior. One such commonly found approach is proposed by Conti and Abdullah. The approach attempts to detect distributed scans against a background of normal traffic based on visualization. Due to lack of details, it is difficult to understand how a distributed scan would use this tool.

VI. RESULTS OF THREAT DETECTION

6.1 PORT SCANNING RESULTS

The result of a scan on a port is usually generalized into one of three categories:

- Open or Accepted: The host sent a reply indicating that a service is listening on the port
- Closed or Denied or not listening: The host sent a reply indicating that connections will be denied to the port.
- Filtered , Dropped or Blocked: There was no reply from the host

Open ports present two vulnerabilities of which administrators must be wary:

- Security and stability concerns associated with the program responsible for delivering the service
- Security and stability concerns associated with the operating system that is running on the host

Closed ports only present the later of the two vulnerabilities that open ports do. Blocked ports do not present any reasonable vulnerability. Of course, there is the possibility that there aren't any known vulnerabilities in either the software or operating system at this given time.

The information gathered by a port scan has many legitimate uses including the ability to verify the security of a network. Port scanning can however be used by those who intend to compromise security. Many exploits rely upon port scans to find open ports and send large quantities of data in an attempt to trigger a condition known as a buffer overflow. Such behavior can compromise the security of a network and the computers therein resulting in the loss or exposure of sensitive information and the ability to do work.

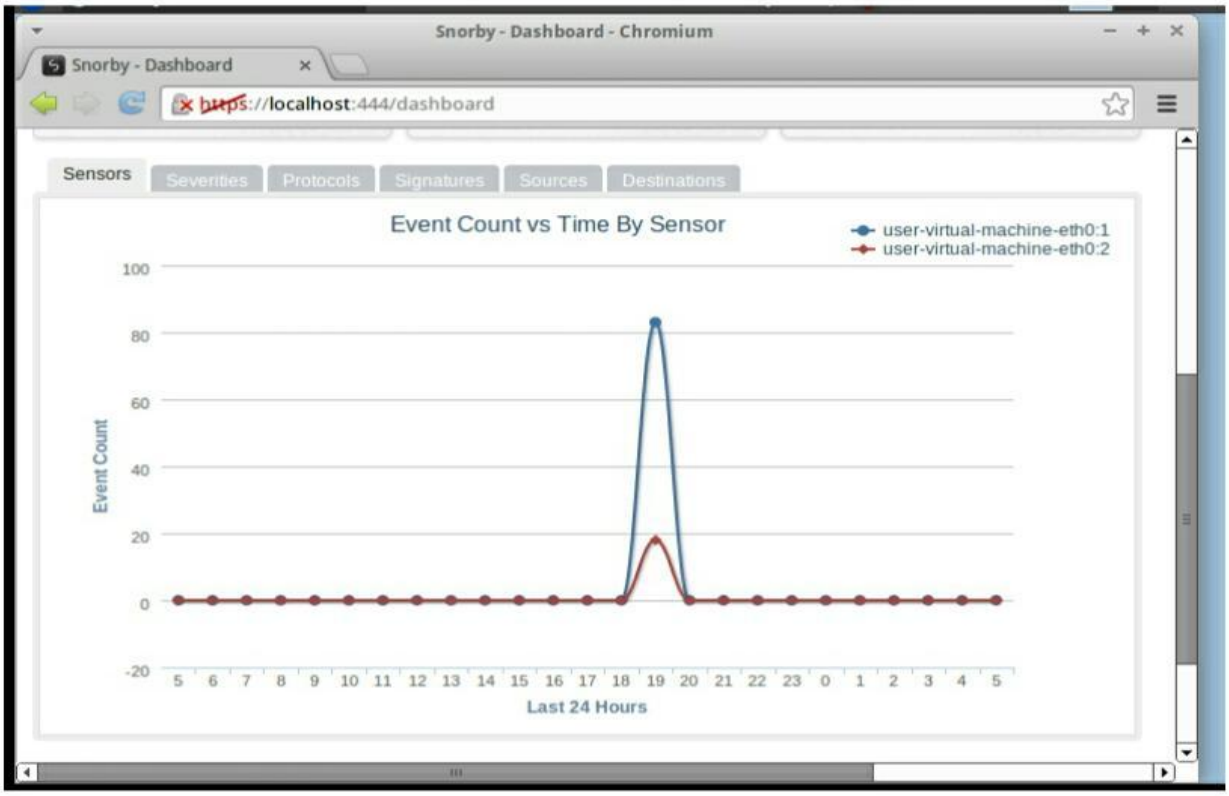

Figure 4. Graphical representation of threat detection

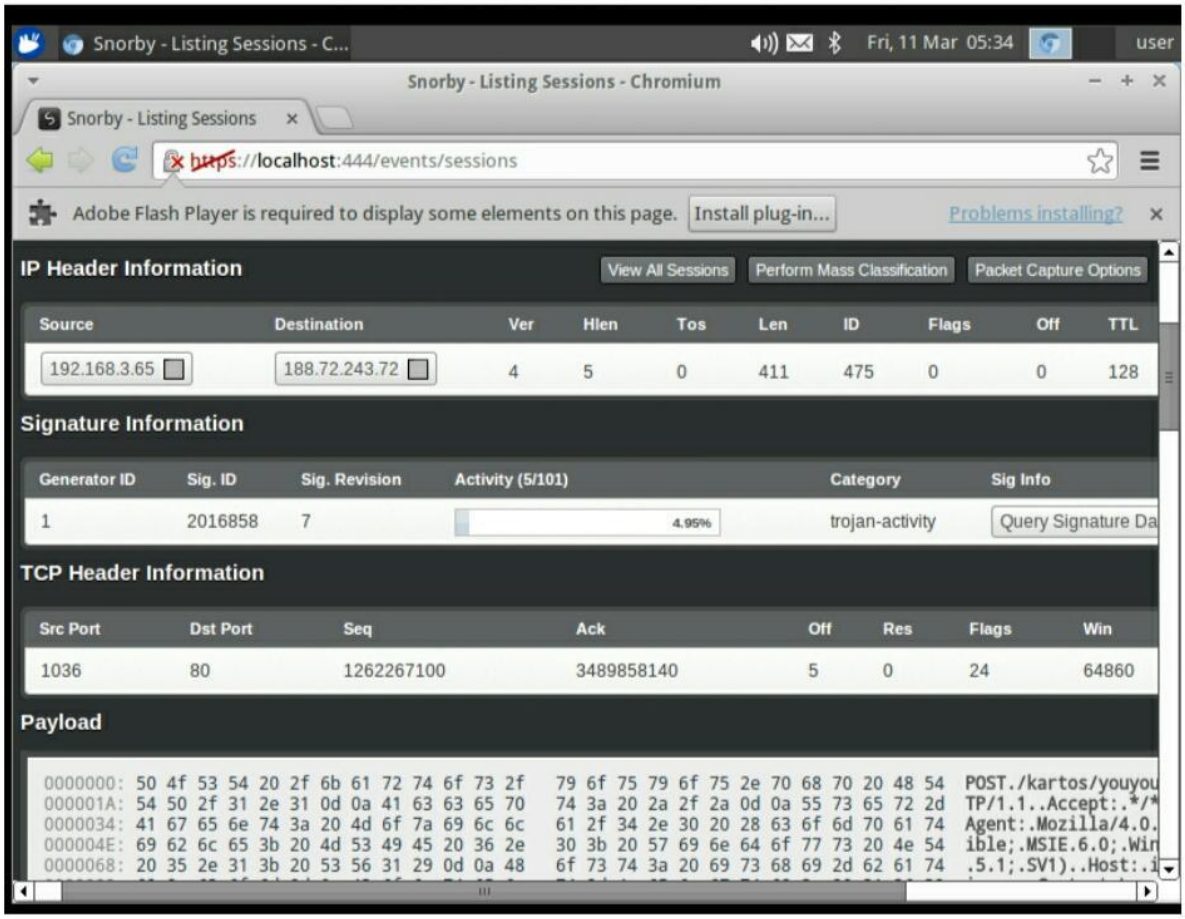

Figure 5. Port scan results

VII. CONCLUSION

In this paper, we have examined the state of modern port scan detection approaches. The discussion follows well-known criteria for categorizing scan detection approaches: detection strategy, data source and data visualization. Experiments demonstrate that for different types of port scan attacks, different anomaly detection schemes may be more successful. Research prototypes combining data mining and threshold based analysis for scan detection have shown great promise. Such detection approaches tend to have lower false positive rates, scalability and robustness. In this existing system we can only scan the ports. We redesigned it by adding additional feature of detecting the threats during scanning itself and securing by not intruding threats. Further, resolving the detected vulnerabilities by additional future development analyzing and testing were done.

REFERENCES

1. A.Alhomoud, R.Munir, J.P.Disso, I.Awan, A. Ai- Dhelaan ,"Performance Evaluation Study of Intrusion Detection System", Procedia Computer Science, vol.5,pp.173-180 on 2011.
2. Z holt, D. Marcelo, Bernardo David, R.T.Sousa, "Building scalable distributed intrusion detection systems ",based on The Map Reduce Framework Telecomunicacoes , Santa Rita do Sapucai:,vol.13 pp. 22-31, 2011.
3. Sachin Shetty, "Auditing and Analysis of Network Traffic in Cloud Environment", 2013 IEEE 9th world Congress on Services, pp 260-267, 2013.
4. Chris Muelder, Kwan-Liu Ma, Tony Bartoletti,"Interactive Visualization for Network and Port Scan Detection".
5. Rashid Munir, Muhammad Rafiq Mufti, Irfan Awan, "Detection, Mitigation and Quantitative Security Risk Assesment of Invisible Attacks at Enterprise Network",Future Internet of Things and cloud ,3rd International Conference on 2015.
6. Travis Boraten, Avinash Karandh K odi, "Mitigation of Denial of Service Attack with hardware Trojans in NoC Architectures", Parallel and Disributed processing symposium, 2016 IEEE International.
7. Jian-wei Tian, HongQiao xili, " A Statistical Threat Detection Method based on Dynamic Time Threshold" in Computer and communications, 2016 2nd IEEE Conference on 11 May 2017.

