Kurundkar G.D, Naik N.A, Dr.Khamitkar S.D / International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 www.ijera.com
Vol. 2, Issue 2,Mar-Apr 2012, pp.1288-1296

Network Intrusion Detection using SNORT

Kurundkar G.D* Naik N.A** Dr.Khamitkar S.D***
*(Dept. of Computer Science SGB College,Purna Dist.Parbhani)
**(Dept. of Computer Science Yeshwant Mahavidyalaya ,Nanded Dist.Nanded)
***(School of Computational Sciences, SRTM University,Nanded(MS)India)

Abstract:

An intruder is a hacker or cracker which always tries to get access to secure, system intrusion occurs when an unauthorized person try to gain access or interrupt the normal operations of an information system. Even when such attacks are self-propagating, as in the case of viruses and distributed denial-of service attacks, they are almost always initiated by an individual whose purpose is to harm an organizational data. Intrusion detection consists of procedures for detection of illegal activity of (intruders) system that identify the intruders. Some important intrusion prevention activities are writing and implementing good activity information security rule, planning and performing effective information security programs, installing and testing technology-based information security system for counting intruders activities such as firewalls and intrusion detection and prevention systems. In Information security intrusion detection systems (IDS) works like a burglar alarm in that it detects destruction and activates an alarm. Recently new technology for IDS systems is the intrusion prevention system (IPS), which can detect an intrusion and also prevent that intrusion from attacking the organization. There is a system called intrusion detection/prevention system (IDPS).Recently Snort is a very useful tool for Network based Intrusion detection. A Snort is tool which can give alert/alarm to the authentic user or Network Administrator by sending email or giving alarm for illegal network activities.

Key Words: Intruder, Prevention, Measurers, attacks, Snort, Activities, Detection, Session,MD5

Need of Intrusion Detection System

When we are working on the Internet it becomes our responsibility make our network more secure by using Network monitoring tools and making security settings and there are several other reasons to use an Intrusion Detection System.

- To detect attacks that are not prevented by other security measures
- To detect and deal with attacks
- To perform as quality organize for security design and administration, especially of large and complex enterprises
- To provide useful information about intrusions that do take place, allowing improved finding, improvement, and correction of contributing factors

1. Network Based Intrusion Detection and Prevention System

A Network Based IDS (NIDS) present in a computer or device connected to a segment of an organization's network and monitors network traffic on that network segment, looking for ongoing attacks. In network for maintain security to files many various Hashing algorithms are used like MD5. When a circumstances occurs that the network-based IDS is planned to know an attack, it responds by sending notifications to administrators. NIDS looks for attack patterns within network traffic, such as large collections of related items that are of a certain type that could specify that a denial-of-service attack is ongoing, or it looks for the exchange of a sequence of related packets in a certain pattern, which could indicate that a port scan is in progress. NIDSs are installed at a specific place in the network e.g router from where it is possible to watch the traffic going into and out of a particular network segment and it can be used to watch specific host computers on a network segment, or it can be installed to monitor all traffic between the systems that make up an entire network. A fundamental problem for network intrusion detection systems (NIDSs) that passively monitor a network link is the ability of a skilled attacker to *evade* detection by exploiting ambiguities in the traffic stream as seen by the NIDS [1].

2. Network Behavior Analysis System

Network Behavior Analysis (NBA) systems examine network traffic in order to identify problems related to the flow of traffic. Network Behavior Analysis, or shortly, was initially designed as a security technology whose purpose is to identify un-usual traffic on the network being supervised [2]. They use a description of the anomaly detection technique described later in this section to identify excessive packet flows such as those that might occur in the case of equipment failure, denial-of-service attacks, virus and worm attacks, and some forms of network policy violations.NBA.

3. Host Based Intrusion Detection System

A Host Based Intrusion Detection System (HIDS) is situated on a particular computer or server, known as the host, and monitors activity only on that system. Host-based intrusion detection systems can be further divided into two categories: signature-based (i.e. misuse detection) and anomaly detection [3]. HIDS monitor the status of key system files and detect when an intruder creates, modifies, or deletes monitored files. The HIDS then triggers an alert when one of the following changes occurs: file attributes change, new files are created, or existing files are deleted. A HIDS has an advantage over NIDS in that it can usually be installed in such a way that it can access information that is encrypted when traveling over the network.

Usefulness of HIDS

- Logs. A HIDS can detect local events on host systems and also detect attacks that may avoid network-based IDS.
- HIDS encrypted traffic will have been decrypted and is available for processing.
- The use of switched network protocols does not affect a HIDS.
- A HIDS can detect inconsistencies in how applications and systems programs were used by examining the records stored in audit

Drawbacks of HIDSs

- A HIDS is vulnerable to some denial-of-service attacks.
- A HIDS can use large amounts of disk space to retain the host OS audit logs, and, to role properly, it may require disk capacity to be added to the system.
- A HIDS can inflict a performance overhead on its host systems, and, in some cases, may reduce system performance below acceptable level

Intrusion Detection Prevention System Methods

IDPS provides multiplicity of detection methods to monitor and calculate approximately network traffic, following are some important methods.

I. Signature Based Intrusion Detection System:

Signature-based detection is normally used for detecting known attacks. A signature-based ID is useful in data traffic which search patterns that match known signatures that is, preconfigured, predetermined attack patterns. Signature-based IDS technology is widely used because many attacks have clear and distinct signatures. In signature-based IDS is that every signature requires an entry in the database, and so a complete database might contain hundreds or even thousands of entries. Each packet is to be compared with all the entries in the database. [4]

II. Statistical Anomaly Based Intrusion Detection System:

In statistical-based techniques, the network traffic activity is captured and a profile representing its stochastic behavior is created. This profile is based on metrics such as the traffic rate, the number of packets for each protocol [5]. Anomaly-based intrusion detection triggers an alarm on the IDS when some type of unusual behavior occurs on your network. This would include any event, state, content, or behavior that is considered to be abnormal by a pre-defined standard [6].

The Statistical Anomaly based IDS or behavior-based IDS collects statistical summaries by observing traffic that is known to be normal. This normal period of evaluation establishes a performance baseline. The data that is measured from the normal traffic and is used to prepare the baseline can include variables such as host memory or CPU usage, network packet types, and packet quantities. The advantage of the statistical anomaly-based approach is that the IDS can detect new types of attacks because it is looking for abnormal activity of any type.

III. Stateful Protocol Analysis Intrusion Detection Prevention System:

Stateful inspection improves on the functions of packet filters by tracking the state of connections and blocking packets that deviate from the expected state. As with packet filtering, stateful inspection intercepts packets at the network layer and inspects them to see if they are permitted by an existing firewall rule, but unlike packet filtering, stateful inspection keeps track of each connection in a state table. While the details of state table entries differ by firewall product, they typically include source IP address, destination IP address, port numbers, and connection state information. [7] Stateful protocol analysis (SPA) is a process of comparing predetermined profiles of generally usual definitions of benign activity for each protocol state against observed events to identify deviations. By storing relevant data detected in a session and then using that data to identify intrusions that involve multiple requests and responses.

Intrusion Detection Prevention System Response

Intrusion prevention systems (IPS), also known as intrusion detection and prevention systems (IDPS), are network security appliances that monitor network and/or system activities for malicious activity. The main functions of intrusion prevention systems are to identify malicious activity, log information about said activity, attempt to block/stop activity, and report activity. [8] Intrusion prevention systems are considered extensions of intrusion detection systems because they both monitor network traffic and/or system activities for malicious activity. The main differences are, unlike intrusion detection systems, intrusion prevention systems are placed in-line and are able to actively prevent/block intrusions that are detected. [9][10] In this IDPSR it has a number of response options, depending on the organization's policy, objectives, and system capabilities.

The following are some of the responses that an IDS.

- alarms
- E-mail messages
- Log entries
- Evidentiary packet dumps
- Take action against the intruder
- Reconfigure firewall
- Terminate session
- Terminate connection

Selecting Intrusion Detection System Products

Now a day's various Intrusion detection products are easily available, according to security goals and organization considerations. It performs variety of features. The process of selecting products that represent the best fit for any specific organization's needs is challenging.

Technical and Policy Considerations

- What is your systems environment?
- What are the technical specifications of your systems environment?
- What are the technical specifications of your current security protections?
- What are the goals of your enterprise?
- What are your security goals and objectives?
- Is your organization concerned about insider attacks?
- Does your organization want to use the output of your IDS to determine new needs?
- Does your organization want to use IDS to maintain managerial control over network usage?
- What is your existing security policy?
- What is the budget for acquisition and life cycle support of intrusion detection hardware, software, and infrastructure?
- Is there sufficient existing staff to monitor an IDS full time?
- Does your organization have authority to instigate changes based on the findings of IDS?

Case Study Snort on Centos:

Cento is an Enterprise-class Linux Distribution derived from sources freely provided to the public by a prominent North American Enterprise Linux vendor. Cento conforms fully to the upstream vendor's redistribution policy. Cento is free. Cento is developed by a small but growing team of core developers. In turn the core developers are supported by an active user community including system administrators, network administrators, enterprise users, managers. Centos has numerous advantages over some of the other clone projects including: an active and growing user community, quickly rebuilt, tested, packages, an extensive mirror network, developers who are contactable and responsive, multiple free support avenues including IRC Chat, Mailing Lists, Forums. Centos overtook Debian to become the most popular Linux distribution for web servers.

Snort:

Snort is an open source network intrusion prevention and detection system utilizing a rule-driven language, which combines the benefits of signature, protocol and anomaly based inspection methods. With millions of downloads to date, Snort is the most widely deployed intrusion detection and prevention technology worldwide and has become the in reality standard for the industry. Snort is used primarily to passively monitor network traffic and generate alerts when threats are detected. More recently, the Inline mode of deployment has become available and can be used to actively intercept and drop network traffic. The essence of Inline mode is that, a) Snort is configured and deployed on a server that forwards/routes network traffic as opposed to only sniffing network traffic and, b) Snort “alert” rules are changed into “drop” rules. Many Linux distributions include the iptables firewall application and Snort Inline interacts with iptables to receive and process network traffic. Appropriate iptables rules are used to direct network traffic to Snort Inline for inspection according to Snort rules. Given this interaction between Snort Inline and iptables, successful configuration of Snort Inline depends on successful configuration of iptables. Accordingly, these notes provide an example of an iptables rule set that supports both integration with Snort Inline and interoperability.

Components of Snort

Snort is logically divided into multiple components. These components work together to detect particular attacks and to generate output in a required format from the detection system. A Snort based IDS consists of the following major components:

- Packet Decoder
- Preprocessors
- Detection Engine
- Logging and Alerting System
- Output Modules

Fig.1.1 shows how these components are arranged. Any data packet coming from the Internet enters the packet decoder. On its way towards the output modules, it is either dropped, logged or an alert is generated.[11]

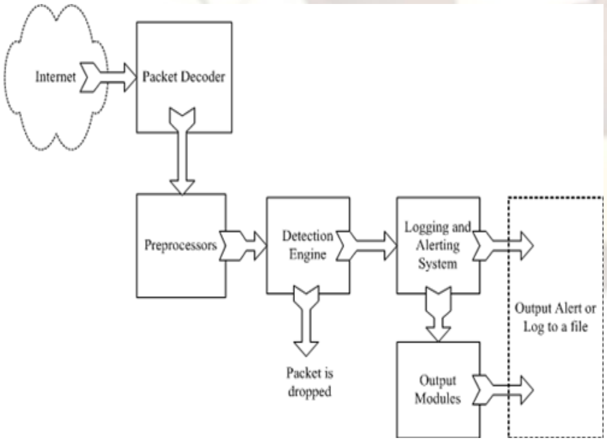


Fig.1.1 Components of Snort.

A brief introduction to these components is presented in this section.

1. Packet Decoder

The packet decoder takes packets from different types of network interfaces and prepares the packets to be preprocessed or to be sent to the detection engine. The interfaces may be Ethernet,
SLIP, PPP and so on.

2. Preprocessors

Preprocessors are components or plug-ins that can be used with Snort to arrange or modify data packets before the detection engine does some operation to find out if the packet is being used by an intruder. Some preprocessors also perform detection by finding anomalies in packet headers and generating alerts. Preprocessors are very important for any IDS to prepare data packets to be

3. The Detection Engine

The detection engine is the most important part of Snort. Its responsibility is to detect if any intrusion activity exists in a packet. The detection engine employs Snort rules for this purpose. The rules are read into internal data structures or chains where they are matched against all packets. If a packet matches any rule, appropriate action is taken; otherwise the packet is dropped. Appropriate actions may be logging the packet or generating alerts.

4. Logging and Alerting System

Depending upon what the detection engine finds inside a packet, the packet may be used to log the activity or generate an alert. Logs are kept in simple text files, tcpdump-style files or some other form. All of the log files are stored under /var/log/snort folder by default. You can use –l command line options to modify the location of generating logs and alerts. Many command line options discussed in the next chapter can modify the type and detail of information that is logged by the logging and alerting system.

5. Output Modules

Output modules or plug-ins can do different operations depending on how you want to save output generated by the logging and alerting system of Snort. Basically these modules control the type of output generated by the logging and alerting system. Depending on the configuration, output modules can do things like the following:

- Simply logging to /var/log/snort/alerts file or some other file
- Sending SNMP traps
- Sending messages to syslog facility
- Logging to a database like MySQL or Oracle. You will learn more about using MySQL later in this book
- Generating extensible Markup Language (XML) output
- Modifying configuration on routers and firewalls.
- Sending Server Message Block (SMB) messages to Microsoft Windows-based machines

How to Protect IDS Itself

One major subject is how to protect the system on which your intrusion detection software is running. If security of the IDS is compromised, you may start getting false alarms or no alarms at all. The intruder may disable IDS before actually performing any attack. There are different ways to protect your system, starting from exceptionally wide-ranging recommendations to some sophisticated methods. Some of these are mentioned below.

- The first obsession that you can do is not to run any service on your IDS sensor itself. Network servers are the most common method of exploiting a system.
- New threats are discovered and patches are released by vendors. This is almost a continuous and non-stop process. The platform on which you are running IDS should be patched with the latest releases. For example, if Snort is running on a Microsoft Windows machine, you should have all the latest security patches from Microsoft installed.
- Configure the IDS machine so that it does not respond to ping (ICMP Echo-type) packets.
- If you are running Snort on a Linux machine, use net filter / iptable to block any unwanted data. Snort will still be able to see all of the data.

Snort with no IP Address Interface

You can also use Snort on an interface where no IP address is assigned. For example, on a Linux machine, you can bring up interface eth0 using command "ifconfig eth0 up" without assigning an actual IP address. The advantage is that when the Snort host doesn't have an IP address itself, nobody can access it. You can configure an IP address on eth1 that can be used to access the sensor itself. This is shown in Fig.1.2.

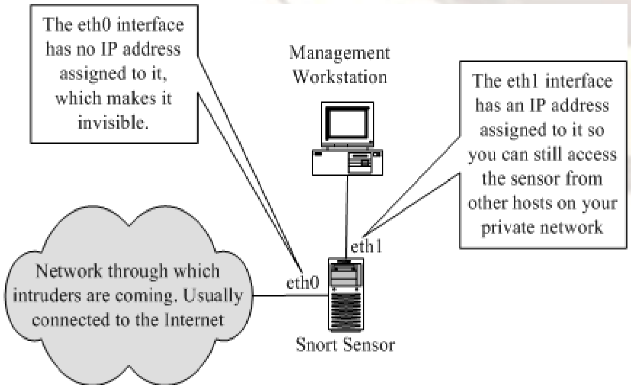

The eth0 interface has no IP address assigned to it, which makes it invisible.
Management Workstation
The eth1 interface has an IP address assigned to it so you can still access the sensor from other hosts on your private network
eth1
eth0
Network through which intruders are coming. Usually connected to the Internet
Snort Sensor

Fig. 1.2 Snort sensors with two interfaces. One of these has no IP address assigned.

On Microsoft Windows systems, you can use an interface without binding TCP/IP to the interface, in which case no IP address will be assigned to the interface. Don't forget to disable other protocols and services on the interface as well. In some cases it has been noted that winpcap (library used on Microsoft Windows machines to capture packets) does not work well when no IP address is assigned on the interface. In such a case, you can use the following method.

- Enable TCP/IP on the network interface that you want to use in the stealth mode. Disable everything other than TCP/IP.
- Enable DHCP client.
- Disable DHCP service.

This will cause no address to be assigned to the interface while the interface is still bound to TCP/IP networking.

Network Intrusion Detection Mode

Intrusion detection is a new retrofit approach for proving a sense of security in existing and data network while allowing them to operate in their current “open” mode.[12] In intrusion detection mode, Snort does not log each captured packet as it does in the network sniffer mode. Instead, it applies rules on all captured packets. If a packet matches a rule, only then is it logged or an alert is generated. If a packet does not match any rule, the packet is dropped silently and no log entry is created. When you use Snort in intrusion detection mode, typically you provide a configuration file on the command line. This configuration file contains Snort rules or reference to other files that contain Snort rules. In addition to rules, the configuration file also contains information about input and output plug-ins.[11][13] The typical name of the Snort configuration file is snort.conf. We have previously saved snort.conf configuration file in /opt/snort/etc directory along with other files. This was done during the installation procedure. The following command starts Snort in the Network Intrusion Detection (NID) mode:

Snort Alert Modes

When Snort is running in the Network Intrusion Detection (NID) mode, it generates alerts when a captured packet matches a rule. Snort can send alerts in many modes. These modes are configurable through the command line as well as through snort.conf file. Common alert modes are explained in this section. To explain the alert modes, I have used a rule that creates an alert when Snort detects an ICMP packet with TTL 100. This rule is listed below.

```
alert icmp any any -> any any (msg: "Ping with TTL=100"; \ttl:100;)
```

Sending Alerts to Syslog

This command allows Snort to send alerts to Syslog daemon. Syslog is a system logger daemon and it generates log files for system events. It reads its configuration file /etc/syslog.conf where the location of these log files is configured. The usual location of syslog files is /var/log directory. On Linux systems, usually /var/log/messages is the main logging file. For more information, use the "man syslog" command. The "man syslog.conf" command shows the format of the syslog.conf file.

Sending Alerts to Windows

Snort can send alerts to Microsoft Windows machines in the form of pop-up windows. These pop-up windows are controlled by Windows Messenger Service. Windows Messenger Service must be running on your Windows machine for pop-up windows to work. You can go to Control Panel and start the *Services* applet to find out if Windows Messenger Service is running. The *Services* applet is found in the Administrative Tools menu on your Windows system. Depending on your version of Microsoft Windows, it may be found in Control Panel or some other place.

Working with Snort Rules

Like viruses, most intruder activity has some sort of signature. Information about these signatures is used to create Snort rules. But we can use honey pots to find out what intruders are doing and information about their tools and techniques. In addition to that, there are databases of known vulnerabilities that intruders want to exploit. These known attacks are also used as signatures to find out if someone is trying to exploit them. These signatures may be present in the header parts of a packet or in the payload. Snort's detection system is based on rules. These rules in turn are based on intruder signatures. Snort rules can be used to check various parts of a data packet. Snort 1.x versions can analyze layer 3 and 4 headers but are not able to analyze application layer protocols. Upcoming Snort version 2 is expected to add support of application layer headers as well. Rules are applied in an orderly fashion to all packets depending on their types. A rule may be used to generate an alert message, log a message, or, in terms of Snort, pass the data packet, i.e., drop it silently. The word pass here is not equivalent to the traditional meaning of pass as used in firewalls and routers. In firewalls and routers, pass and drop are opposite to each other. Snort rules are written in an easy to understand syntax. Most of the rules are written in a single line. However you can also extend rules to multiple lines by using a backslash character at the end of lines. Rules are usually placed in a configuration file, typically snort.conf You can also use multiple files by including them in a main configuration file. This chapter provides information about different types of rules as well as the basic structure of a rule. You will find many examples of common rules for intrusion detection activity at the end of this chapter. After reading this chapter, along with the two preceding chapters, you should have enough information to set up Snort as a basic intrusion detection system.

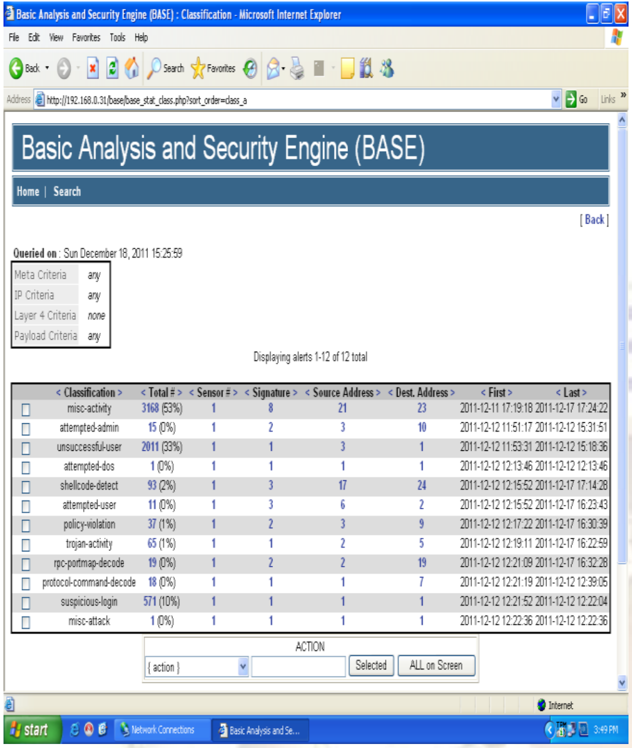
User Defined Actions

These rule actions can be used for different purposes, such as:

- Sending messages to syslog. Syslog is system logger daemon and creates log file in /var/log directory. Location of these files can be changed using /etc/syslog.conf file. For more information, use "man syslog" and "man syslog.conf" commands on a UNIX system. Syslog may be compared to the event logger on Microsoft Windows systems.

Port Number

The port number is used to apply a rule on packets that originate from or go to a particular port or a range of ports. For example, you can use source port number 23 to apply a rule to those packets that originate from a Telnet server. You can use the keyword *any* to apply the rule on all packets irrespective of the port number. Port number is meaningful only for TCP and UDP protocols. If you have selected IP or ICMP as the protocol in the rule, port number does not play any role. The following rule is applied to all packets that originate from a Telnet server in 192.168.2.0/24, which is a class C network and contains the word "confidential": alert tcp 192.168.2.0/24 23 -> any any \ (content: "confidential"; msg: "Detected confidential";)
To detect this type of TCP ping, you can have a rule like the following that sends an alert message: alert tcp any any -> 192.168.1.0/24 any (flags: A; \ ack: 0; msg: "TCP ping detected";)


Basic Analysis and Security Engine (BASE) : Classification - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://192.168.0.31/base/base_stat_class.php?sort_order=class_a
Go Links
Basic Analysis and Security Engine (BASE)
Home | Search
[Back]
Internet
start Network Connections Basic Analysis and Se... 3:49 PM

Queried on : Sun December 18, 2011 15:25:59

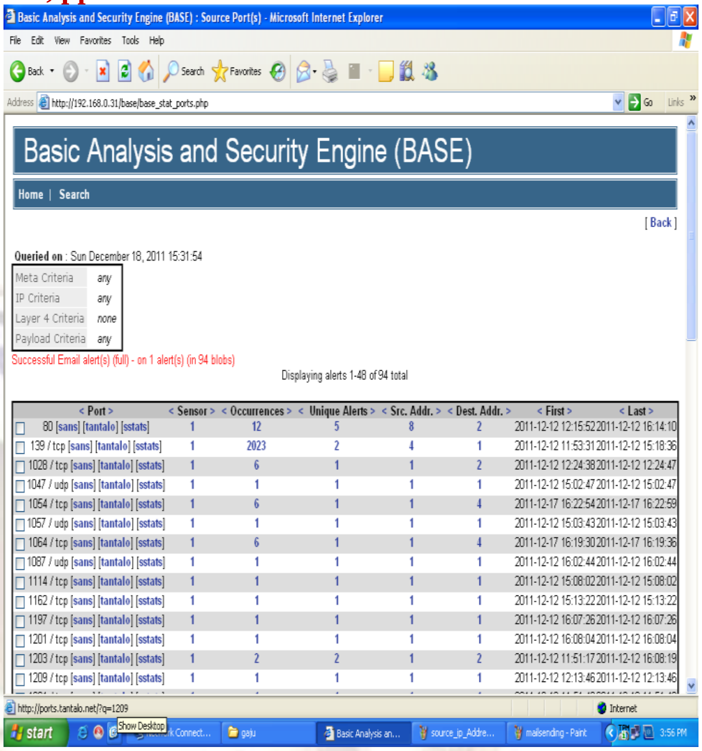
Meta Criteria	any
IP Criteria	any
Layer 4 Criteria	none
Payload Criteria	any

Displaying alerts 1-12 of 12 total

	< Classification >	< Total # >	< Sensor # >	< Signature >	< Source Address >	< Dest. Address >	< First >	< Last >
□	misc-activity	3168 (53%)	1	8	21	23	2011-12-11 17:19:18	2011-12-17 17:24:22
□	attempted-admin	15 (0%)	1	2	3	10	2011-12-12 11:51:17	2011-12-12 15:31:51
□	unsuccessful-user	2011 (33%)	1	1	3	1	2011-12-12 11:53:31	2011-12-12 15:18:36
□	attempted-dos	1 (0%)	1	1	1	1	2011-12-12 12:13:46	2011-12-12 12:13:46
□	shellcode-detect	93 (2%)	1	3	17	24	2011-12-12 12:15:52	2011-12-17 17:14:28
□	attempted-user	11 (0%)	1	3	6	2	2011-12-12 12:15:52	2011-12-17 16:23:43
□	policy-violation	37 (1%)	1	2	3	9	2011-12-12 12:17:22	2011-12-17 16:30:39
□	trojan-activity	65 (1%)	1	1	2	5	2011-12-12 12:19:11	2011-12-17 16:22:59
□	rpc-portmap-decode	19 (0%)	1	2	2	19	2011-12-12 12:21:09	2011-12-17 16:32:28
□	protocol-command-decode	18 (0%)	1	1	1	7	2011-12-12 12:21:19	2011-12-12 12:39:05
□	suspicious-login	571 (10%)	1	1	1	1	2011-12-12 12:21:52	2011-12-12 12:22:04
□	misc-attack	1 (0%)	1	1	1	1	2011-12-12 12:22:36	2011-12-12 12:22:36

ACTION
{ action } | Selected | ALL on Screen

Figure 1.3. SNORT home page Displaying Snort alerts inside BASE window.


Basic Analysis and Security Engine (BASE) : Source Port(s) - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://192.168.0.31/base/base_stat_ports.php
Go Links
Basic Analysis and Security Engine (BASE)
Home | Search
[Back]
http://ports.tantalo.net/?q=1209
Internet
start Show Desktop Network Connect... gaju Basic Analysis an... source_ip_Addre... mailsending - Paint 3:56 PM

Queried on : Sun December 18, 2011 15:31:54

Meta Criteria	any
IP Criteria	any
Layer 4 Criteria	none
Payload Criteria	any

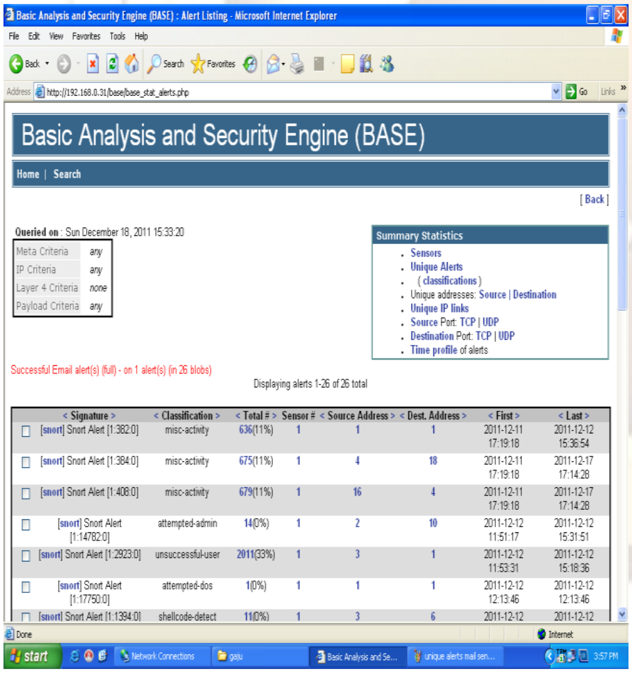
Successful Email alert(s) (full) - on 1 alert(s) (in 94 blobs)

Displaying alerts 1-48 of 94 total

	< Port >	< Sensor >	< Occurrences >	< Unique Alerts >	< Src. Addr. >	< Dest. Addr. >	< First >	< Last >
□	80 [sans] [tantalo] [sstats]	1	12	5	8	2	2011-12-12 12:15:52	2011-12-12 16:14:10
□	139 / tcp [sans] [tantalo] [sstats]	1	2023	2	4	1	2011-12-12 11:53:31	2011-12-12 15:18:36
□	1028 / tcp [sans] [tantalo] [sstats]	1	6	1	1	2	2011-12-12 12:24:38	2011-12-12 12:24:47
□	1047 / udp [sans] [tantalo] [sstats]	1	1	1	1	1	2011-12-12 15:02:47	2011-12-12 15:02:47
□	1054 / tcp [sans] [tantalo] [sstats]	1	6	1	1	4	2011-12-17 16:22:54	2011-12-17 16:22:59
□	1057 / udp [sans] [tantalo] [sstats]	1	1	1	1	1	2011-12-12 15:03:43	2011-12-12 15:03:43
□	1064 / tcp [sans] [tantalo] [sstats]	1	6	1	1	4	2011-12-17 16:19:30	2011-12-17 16:19:36
□	1087 / udp [sans] [tantalo] [sstats]	1	1	1	1	1	2011-12-12 16:02:44	2011-12-12 16:02:44
□	1114 / tcp [sans] [tantalo] [sstats]	1	1	1	1	1	2011-12-12 15:08:02	2011-12-12 15:08:02
□	1162 / tcp [sans] [tantalo] [sstats]	1	1	1	1	1	2011-12-12 15:13:22	2011-12-12 15:13:22
□	1197 / tcp [sans] [tantalo] [sstats]	1	1	1	1	1	2011-12-12 16:07:26	2011-12-12 16:07:26
□	1201 / tcp [sans] [tantalo] [sstats]	1	1	1	1	1	2011-12-12 16:08:04	2011-12-12 16:08:04
□	1203 / tcp [sans] [tantalo] [sstats]	1	2	2	1	2	2011-12-12 11:51:17	2011-12-12 16:08:19
□	1209 / tcp [sans] [tantalo] [sstats]	1	1	1	1	1	2011-12-12 12:13:46	2011-12-12 12:13:46

Fig. 1.4 and 1.5 Last 24 hours listing alerts through E-mail on SNORT BASE

-


Basic Analysis and Security Engine (BASE) : Alert Listing - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://192.168.0.31/base/base_stat_alerts.php
Go Links
Basic Analysis and Security Engine (BASE)
Home | Search
[Back]
Done
Internet
start Network Connections gaju Basic Analysis and Se... unique alerts mail sen... 3:57 PM

Queried on : Sun December 18, 2011 15:33:20

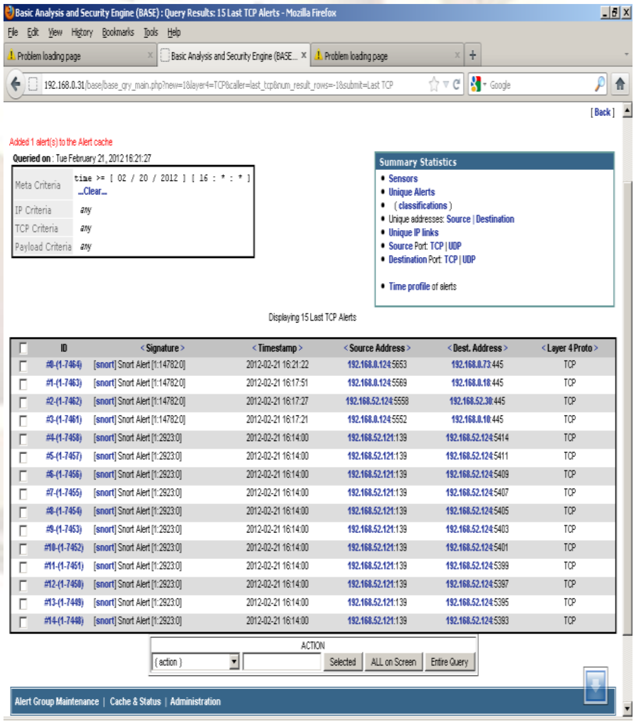
Meta Criteria	any
IP Criteria	any
Layer 4 Criteria	none
Payload Criteria	any

Summary Statistics
- Sensors
- Unique Alerts
- (classifications)
- Unique addresses: Source | Destination
- Unique IP links
- Source Port: TCP | UDP
- Destination Port: TCP | UDP
- Time profile of alerts

Successful Email alert(s) (full) - on 1 alert(s) (in 26 blobs)

Displaying alerts 1-26 of 26 total

	< Signature >	< Classification >	< Total # >	Sensor #	< Source Address >	< Dest. Address >	< First >	< Last >
□	[snort] Snort Alert [1:382:0]	misc-activity	636(11%)	1	1	1	2011-12-11 17:19:18	2011-12-12 15:36:54
□	[snort] Snort Alert [1:384:0]	misc-activity	675(11%)	1	4	18	2011-12-11 17:19:18	2011-12-17 17:14:28
□	[snort] Snort Alert [1:408:0]	misc-activity	679(11%)	1	16	4	2011-12-11 17:19:18	2011-12-17 17:14:28
□	[snort] Snort Alert [1:14782:0]	attempted-admin	14(0%)	1	2	10	2011-12-12 11:51:17	2011-12-12 15:31:51
□	[snort] Snort Alert [1:2923:0]	unsuccessful-user	2011(33%)	1	3	1	2011-12-12 11:53:31	2011-12-12 15:18:36
□	[snort] Snort Alert [1:17750:0]	attempted-dos	1(0%)	1	1	1	2011-12-12 12:13:46	2011-12-12 12:13:46
□	[snort] Snort Alert [1:1394:0]	shellcode-detect	11(0%)	1	3	6	2011-12-12	2011-12-12


Basic Analysis and Security Engine (BASE) : Query Results: 15 Last TCP Alerts - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Problem loading page
Basic Analysis and Security Engine (BASE...
Problem loading page
192.168.0.31/base/base_qry_main.php?new=1&layer4=TCP&caller=last_tcp&num_result_rows=-1&submit=Last TCP
Google
[Back]
Alert Group Maintenance | Cache & Status | Administration

Added 1 alert(s) to the Alert cache

Queried on : Tue February 21, 2012 16:21:27

Meta Criteria	time >= [02 / 20 / 2012] [16 : * : *] ...Clear...
IP Criteria	any
TCP Criteria	any
Payload Criteria	any

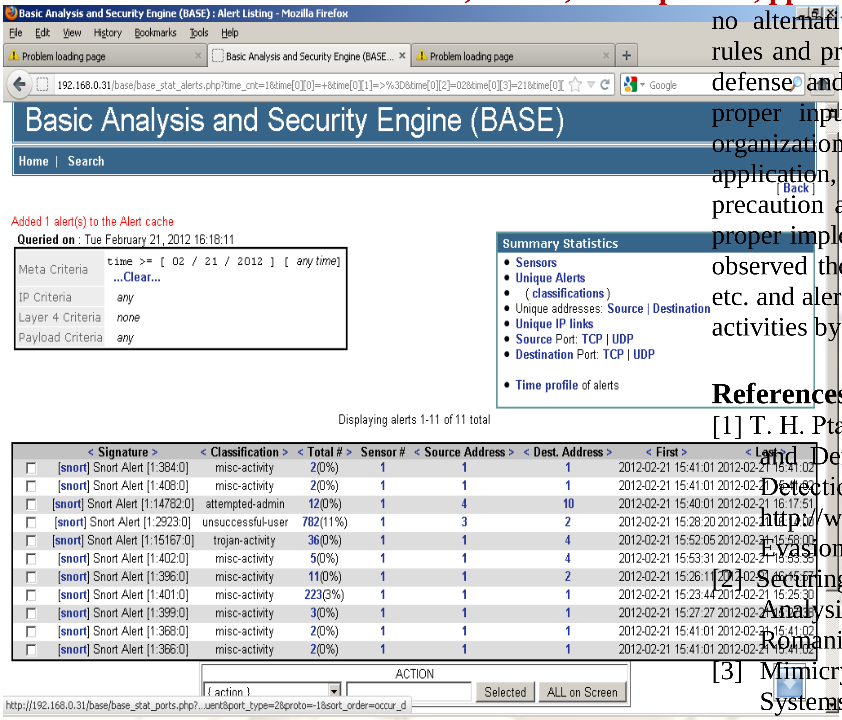
Summary Statistics
- Sensors
- Unique Alerts
- (classifications)
- Unique addresses: Source | Destination
- Unique IP links
- Source Port: TCP | UDP
- Destination Port: TCP | UDP
- Time profile of alerts

Displaying 15 Last TCP Alerts

□	ID	< Signature >	< Timestamp >	< Source Address >	< Dest. Address >	< Layer 4 Proto >
□	#0-(1-7464)	[snort] Snort Alert [1:14782:0]	2012-02-21 16:21:22	192.168.0.124:5653	192.168.0.73:445	TCP
□	#1-(1-7463)	[snort] Snort Alert [1:14782:0]	2012-02-21 16:17:51	192.168.0.124:5569	192.168.0.18:445	TCP
□	#2-(1-7462)	[snort] Snort Alert [1:14782:0]	2012-02-21 16:17:27	192.168.52.124:5558	192.168.52.30:445	TCP
□	#3-(1-7461)	[snort] Snort Alert [1:14782:0]	2012-02-21 16:17:21	192.168.0.124:5552	192.168.0.10:445	TCP
□	#4-(1-7458)	[snort] Snort Alert [1:2923:0]	2012-02-21 16:14:00	192.168.52.121:139	192.168.52.124:5414	TCP
□	#5-(1-7457)	[snort] Snort Alert [1:2923:0]	2012-02-21 16:14:00	192.168.52.121:139	192.168.52.124:5411	TCP
□	#6-(1-7456)	[snort] Snort Alert [1:2923:0]	2012-02-21 16:14:00	192.168.52.121:139	192.168.52.124:5409	TCP
□	#7-(1-7455)	[snort] Snort Alert [1:2923:0]	2012-02-21 16:14:00	192.168.52.121:139	192.168.52.124:5407	TCP
□	#8-(1-7454)	[snort] Snort Alert [1:2923:0]	2012-02-21 16:14:00	192.168.52.121:139	192.168.52.124:5405	TCP
□	#9-(1-7453)	[snort] Snort Alert [1:2923:0]	2012-02-21 16:14:00	192.168.52.121:139	192.168.52.124:5403	TCP
□	#10-(1-7452)	[snort] Snort Alert [1:2923:0]	2012-02-21 16:14:00	192.168.52.121:139	192.168.52.124:5401	TCP
□	#11-(1-7451)	[snort] Snort Alert [1:2923:0]	2012-02-21 16:14:00	192.168.52.121:139	192.168.52.124:5399	TCP
□	#12-(1-7450)	[snort] Snort Alert [1:2923:0]	2012-02-21 16:14:00	192.168.52.121:139	192.168.52.124:5397	TCP
□	#13-(1-7449)	[snort] Snort Alert [1:2923:0]	2012-02-21 16:14:00	192.168.52.121:139	192.168.52.124:5395	TCP
□	#14-(1-7448)	[snort] Snort Alert [1:2923:0]	2012-02-21 16:14:00	192.168.52.121:139	192.168.52.124:5393	TCP

ACTION
{ action } | Selected | ALL on Screen | Entire Query

Fig. 1.6 Most Recent 15 alerts TCP on SNORT BASE


Basic Analysis and Security Engine (BASE) : Alert Listing - Mozilla Firefox
File
Edit
View
History
Bookmarks
Tools
Help
Problem loading page
Basic Analysis and Security Engine (BASE...
Problem loading page
192.168.0.31/base/base_stat_alerts.php?time_cnt=1&time[0][0]=+&time[0][1]=>%3D&time[0][2]=02&time[0][3]=21&time[0][
Google
Basic Analysis and Security Engine (BASE)
Home | Search
[Back]

Added 1 alert(s) to the Alert cache

Queried on : Tue February 21, 2012 16:18:11

Meta Criteria	time >= [02 / 21 / 2012] [*any time*] ...Clear...
IP Criteria	*any*
Layer 4 Criteria	*none*
Payload Criteria	*any*

Summary Statistics

- Sensors
- Unique Alerts
- (classifications)
- Unique addresses: Source | Destination
- Unique IP links
- Source Port: TCP | UDP
- Destination Port: TCP | UDP
- Time profile of alerts

Displaying alerts 1-11 of 11 total

	< Signature >	< Classification >	< Total # >	Sensor #	< Source Address >	< Dest. Address >	< First >	< Last >
□	[snort] Snort Alert [1:384:0]	misc-activity	2(0%)	1	1	1	2012-02-21 15:41:01	2012-02-21 15:41:02
□	[snort] Snort Alert [1:408:0]	misc-activity	2(0%)	1	1	1	2012-02-21 15:41:01	2012-02-21 [illegible]
□	[snort] Snort Alert [1:14782:0]	attempted-admin	12(0%)	1	4	10	2012-02-21 15:40:01	2012-02-21 16:17:51
□	[snort] Snort Alert [1:2923:0]	unsuccessful-user	782(11%)	1	3	2	2012-02-21 15:28:20	2012-02-21 [illegible]
□	[snort] Snort Alert [1:15167:0]	trojan-activity	36(0%)	1	1	4	2012-02-21 15:52:05	2012-02-21 15:58:00
□	[snort] Snort Alert [1:402:0]	misc-activity	5(0%)	1	1	4	2012-02-21 15:53:31	2012-02-21 15:53:35
□	[snort] Snort Alert [1:396:0]	misc-activity	11(0%)	1	1	2	2012-02-21 15:26:11	2012-02-21 16:15:57
□	[snort] Snort Alert [1:401:0]	misc-activity	223(3%)	1	1	1	2012-02-21 15:23:44	2012-02-21 15:25:30
□	[snort] Snort Alert [1:399:0]	misc-activity	3(0%)	1	1	1	2012-02-21 15:27:27	2012-02-21 [illegible]
□	[snort] Snort Alert [1:368:0]	misc-activity	2(0%)	1	1	1	2012-02-21 15:41:01	2012-02-21 15:41:02
□	[snort] Snort Alert [1:366:0]	misc-activity	2(0%)	1	1	1	2012-02-21 15:41:01	2012-02-21 15:41:02

http://192.168.0.31/base/base_stat_ports.php?...uent&port_type=2&proto=-1&sort_order=occur_d

Fig. 1.7 Today's Unique Alerts

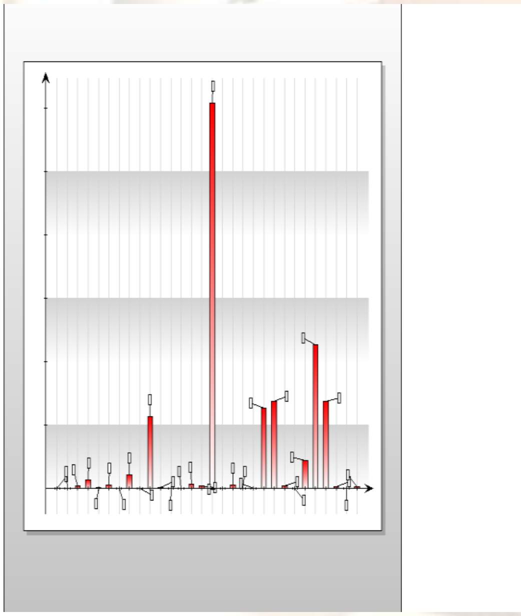


Fig. 1.8 Graphical Representation

Conclusion

Above paper discuss Intrusion Detection Systems and Intrusion Detection and Prevention systems by using freeware Software SNORT tool which can work as web application. But it is important to understand that application based vulnerabilities are different in each application and cannot be resolved by any generic rule which and is possible in network security. Also, there is no alternative of secure coding; adding such generic rules and protecting application is just one more line of defense and it cannot be considered as alternative of proper input validation. The best approach for any organization is perform penetration testing for application, write SNORT rules to protect temporary precaution against attacks and start modifying code for proper implementation of security. We have studied and observed the attacks on different ports like TCP , UDP etc. and alert the administrator via email about the illegal activities by the intruder in home network.

References

[1] T. H. Ptacek and T. N. Newsham, "Insertion, Evasion and Denial of Service: Eluding Network Intrusion Detection", Secure Networks, Inc., Jan. 1998. http://www.aciri.org/vern/Ptacek- Newsham-Evasion-98.ps

[2] Securing the Organization with Network Behavior Analysis by Jack TIMOFTE Praktiker Romania *Economy Informatics*, 1-4/2007 PP.73-76.

[3] Mimicry Attacks on HostBased Intrusion Detection Systems by David Wagner & Paolo Soto *CCS'02,* November 18–22, 2002, Washington, DC, USA.

[4] Dynamic Multi-Layer Signature Based Intrusion Detection System Using Mobile Agents by Mueen Uddin1, Kamran Khowaja2 and Azizah Abdul Rehman in International Journal of Network Security & Its Applications (IJNSA), Vol.2, No.4, October 2010 PP.129-141.

[5] Anomaly-based network intrusion detection: Techniques, systems and challenges by P. Garcı´a-Teodoro , J. Dı´az-Verdejo, G. Macia´-Ferna´ndez & E. Va´zquez in computers & security 28 (2009) Elsevier PP.18-28

[6] Deciphering Detection Techniques: Part II Anomaly-Based Intrusion Detection By Dr. Fengmin Gong, Chief Scientist, McAfee Network Security Technologies Group Network Associates Your Netwok, our business March 2003.

[7] Guidelines on Firewalls and Firewall Policy by Karen Scarfone Paul Hoffman National Institute of standards and Technology sep-2009

[8] NIST – Guide to Intrusion Detection and Prevention Systems (IDPS). 2007-02. Retrieved 2010-06-25.

[9] Robert C. Newman (19 February 2009). *Computer Security: Protecting Digital Resources*. Jones & Bartlett Learning. pp. 273–. ISBN 9780763759940. Retrieved 25 June 2010.

[10] Michael E. Whitman; Herbert J. Mattord (2009). *Principles of Information Security*. Cengage Learning EMEA. pp. 289–. ISBN 9781423901778. Retrieved 25 June 2010.

[11] Intrusion Detection with SNORT: Advanced IDS Techniques Using SNORT, Apache, MySQL, PHP, and ACID by Rafeeq Ur Rehman.

[12] Network intrusion Detection by Biswanath Mukherjee,L.Todd Heberlein and Karl N.Levitt in IEEE Network –may/june-1194. PP.26-41

[13] Intrusion Detection with SNORT: Advanced IDS Techniques Using SNORT, Apache, MySQL, PHP, and ACID by Rafeeq Ur Rehman.

[14] Snort web site at http://www.snort.org