SnortCM: AN APPROACH TO CENTRALIZED INTRUSION DETECTION MANAGEMENT

Alan Christianson
Computer Science
South Dakota School of Mines and Technology
alan@snortcm.com

Nick Rogness
Computer Science
South Dakota School of Mines and Technology
nick@snortcm.com

Wesley Roth
Computer Science
South Dakota School of Mines and Technology
bubba@snortcm.com

John Walton
Computer Science
South Dakota School of Mines and Technology
john@snortcm.com

Dr. Manuel Penaloza
Computer Science
South Dakota School of Mines and Technology
Manuel.Penaloza@sdsmt.edu

Abstract

As the Internet grows, the demand for security related products become more important every day. Several products have recently become available to aid security professionals in tracking, analyzing, and preventing attacks. Among the more popular of these security tools is an intrusion detection system (IDS) engine known as Snort™. Although Snort™ is a powerful application, its lack of a graphical user interface (GUI) results in a high learning curve. In addition, very few centralized management applications exist which interface with Snort™. In an attempt to alleviate these shortcomings, we developed SnortCM.

SnortCM aggregates IDS data from multiple remote Snort™ sensors. SnortCM includes a number of features which allow users to manage Snort™ IDS sensors. Therefore, the use of SnortCM decreases the difficulty of deploying a large number of remote Snort™ sensors across the enterprise.

Introduction

Only a decade ago, personnel with just a basic understanding of security maintained what was then the backbone of the Internet. Although this knowledge didn't prevent break-ins, the number of security breeches remained low. The truth is the Internet is not even moderately secure. Computer break-ins, worms, and email viruses are common front-page news stories and keep network administrators busy.

In 2002, the FBI reported that the cost of online computer crime to corporations totaled over 15 billion dollars [1]. Attacks and security breaches are devastating to companies, forcing them to recognize the need for security in their infrastructure.

Computer incidents are on the rise every year. CERT/CC, the first computer security incident response team, reported 137,439 incidents in 2003, up 60% from 2002 [2]. This is compounded by the fact that tools are readily available, making it easy for anyone with basic computer knowledge to violate people's privacy, steal information, and commit computer crimes.

As a result, security-related products have become a multi-billion dollar industry. Several products have recently become available to aid security professionals in tracking, analyzing, and preventing attacks and security breaches. Among the more popular of these security tools are intrusion detection systems (IDS).

Intrusion Detection System (IDS)

Intrusion detection systems (IDS) monitor network traffic and host audit logs in order to determine whether any violations have taken place. By monitoring activity as it occurs, "…the IDS can identify suspicious behavioral patterns and either notify network administrators, initiate an automated response to the perceived attack, or both." [3]. As the frequency and severity of computer attacks continue to increase, intrusion detection systems will remain an integral part of any security toolkit.

Importance of IDS

By analogy, intrusion detection systems act as the burglar alarms within the network and aid in deterrence, detection, damage assessment, and evidence gathering. In the world of network security, the ability to know when an intruder is engaged in reconnaissance or other malicious activity can mean the difference between being compromised and being protected. Without an IDS facility in place to monitor network and host activity, both attempted and successful intrusion attempts may go unnoticed, resulting in costly or even irreparable damage to an organization's network.

Any entity with a presence on the Internet should have some form of an IDS running as a key line of defense. Furthermore, IDS can detect and deal with insider attacks as well as

external attacks. This makes IDS one of the only proactive means of detecting and responding to internal and external threats.

Intrusion detection systems can issue alerts and take various kinds of automatic action, ranging from shutting down Internet links to launching backtraces or other active attempts to identifying attackers. Different types of IDS exist to detect nefarious actions.

IDS Types

There are two basic types of intrusion detection systems: Host-based and Network-based. Each has a distinct approach to monitoring and securing data.

Host-based IDS (HIDS)

Host-based systems were the first type of IDS to be developed and implemented. Unlike network-based IDS, these systems can detect attacks against a host made by an intruder who is logged in at the host's terminal. A host-based IDS can review the system and event logs in order to detect an attack on the host and determine whether the attack was successful.

Network-based IDS (NIDS)

A network-based IDS usually provides reliable, real-time information without consuming network or host resources. A network-based IDS is passive when acquiring data. Because a network-based IDS reviews packets and headers, it can detect a variety of attacks. Furthermore, because of real-time monitoring, it can respond to an attack in progress to limit damage. One of the most popular NIDS engines is known as Snort ™.

IDS Detection Methods

In addition to the two types noted above, intrusion detection systems can detect attacks through two mechanisms: knowledge-based or behavior-based. These are known as signature-based IDS or statistical anomaly-based IDS, respectively.

Signature-Based

In a signature-based IDS signatures or attributes (which characterize an attack) are stored in the attack signatures database. When data about events are acquired from the host audit logs or from network packet monitoring, this data is compared with the signatures in the database. If a match is found, a response is initiated.

Statistical Anomaly-Based

With this method an IDS acquires data and defines a baseline usage profile for the network or host that is being monitored. This characterization is accomplished by taking statistical samples of the system over a set period of time. Typical characterization information used to establish a normal profile includes memory usage, CPU utilization, and network packet types.

Snort™

Snort™ is an open source network-based intrusion detection systems (NIDS) capable of performing real-time traffic analysis and packet logging on IP networks. It can perform protocol analysis, content searching/matching and can be used to detect a variety of attacks. In addition to its detection engine, which utilizes a modular plug-in architecture, Snort™ uses a flexible rules language to describe traffic that it should collect or pass. Snort™ has a real-time alerting capability incorporating several alerting mechanisms [4].

With industry's widespread adoption and integration of intrusion detection systems such as Snort™, it has become clear that intrusion detection systems are an important part of an organization's infrastructure. Many companies have deployed, or are in the process of deploying, enterprise-wide IDS solutions. As they begin to setup and subsequently administer these systems, companies are experiencing numerous obstacles related to deployment, management, and data collection. Snort™ is no exception. Although it is generally considered the most powerful and widely used IDS, Snort™ often receives criticism for its high learning curve. This is compounded by the fact that there are few graphical user interfaces (GUI) for managing multiple Snort™ sensors.

SnortCM

SnortCM was proposed, designed, and implemented by the student authors of this paper, under the technical supervision of Dr. Manuel Penaloza. These students are currently enrolled in Senior Design at South Dakota School of Mines and Technology. The implementation of SnortCM satisfies one of the requirements for this course.

SnortCM is designed to administer a large network of Snort™ sensors within a single, web-based interface. Furthermore, SnortCM aggregates Snort™ alerts and information on potential attacks. This information is presented to the user in an easy to understand format to aid in incident analysis across the network.

Remote Snort™ IDS sensors report to SnortCM (the centralized management system). Attack logs are periodically uploaded to SnortCM and can be stored in a central database; new attack signatures can be downloaded to the sensors on an as-needed basis. The rules

for each sensor can be tailored to meet its individual needs and alerts can be used to notify the IDS administrator.

SnortCM Overview

The SnortCM application includes a number of features which allow users to manage Snort IDS sensors. SnortCM is logically and programmatically divided into three parts:

1.) User Interface component
2.) Server component
3.) Client component

These components are integrated as shown in Figure 1.

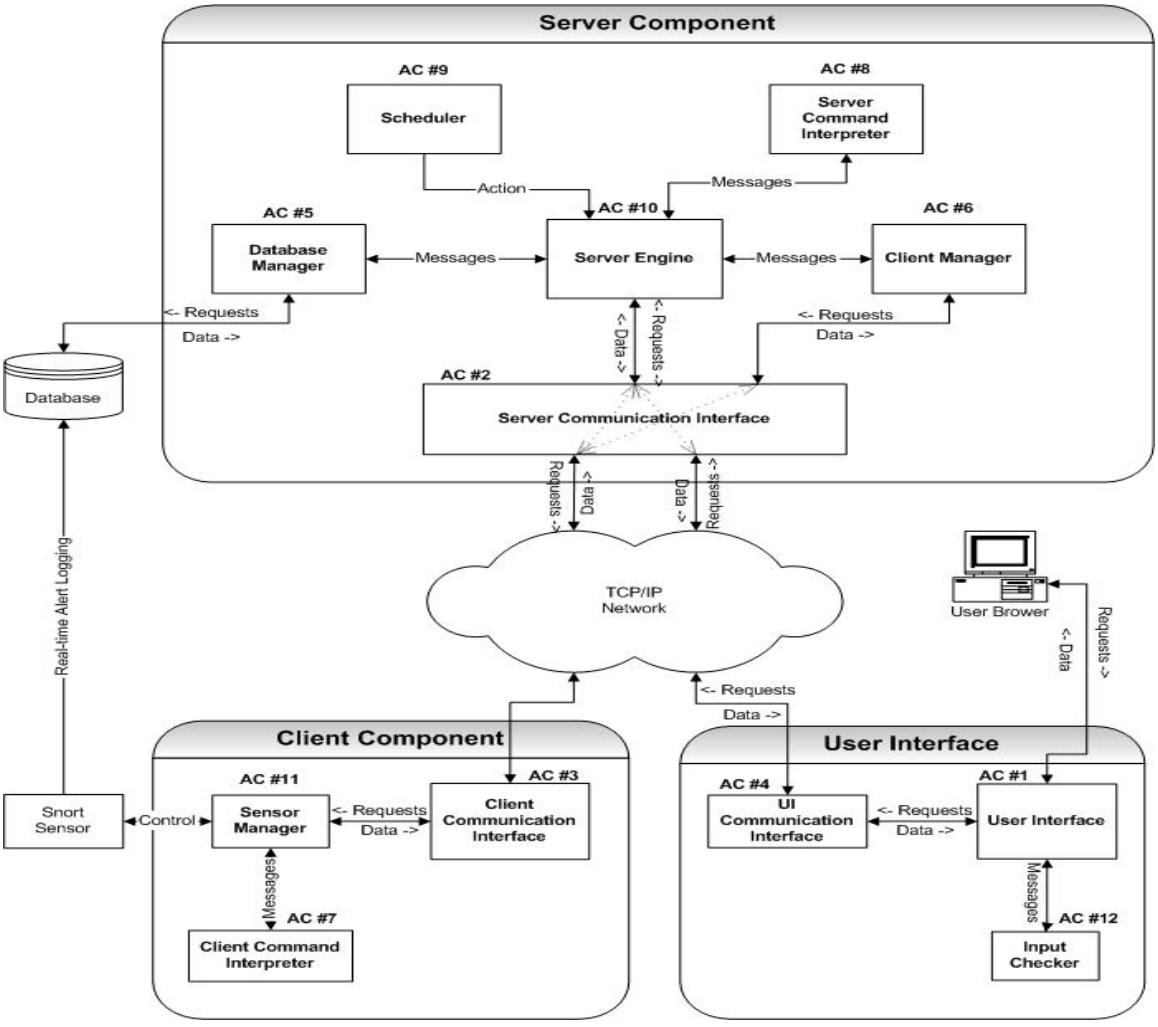

Figure 1: SnortCM Architectural Components

SnortCM Server

The SnortCM server component is the cornerstone of SnortCM. It is responsible for:

- Interpreting commands from the User Interface
- Scheduling events to be processed
- Managing the database
- Controlling the client components
- Serving as the interface between the client and User Interface

SnortCM Client

The SnortCM client component manages the Snort™ application on remote sensors. Additionally, the client reports attack information to the central SnortCM server. The use of multiple SnortCM clients provides the incident analysis team a broader view of the network than can be achieved with single IDS systems. The clients can also be distributed across multiple physical locations, allowing for a single incident analysis team to view attack data across multiple corporate locations.

SnortCM User Interface

This web interface allows users to manage the Snort sensors and summarize pertinent information about attacks. This allows the interactive querying of attack data for analysis. A major feature of the User Interface is that it allows users to control the Snort™ process on a remote system, which includes starting, stopping and polling the system. The User Interface is accessible through any Internet connection using a standard web browser.

SnortCM Implementation

As a product designed for security professionals, security was a priority during the development of SnortCM. For example, throughout the initial design security goals and requirements were defined in documentation. During implementation, these guidelines were followed.

One of the design security goals was dual authentication and encrypted communication between the server and client. In order to satisfy this security goal, SnortCM uses the Secure Socket Layer (SSL) protocol to encrypt all client and server communication. This allows SnortCM to ensure confidentiality and integrity from eavesdroppers and additionally satisfies the design's dual authentication requirement.

Dual authentication between the server and client are performed through the use of X.509 certificates. Each SnortCM client generates a certificate which the server validates. Similar validation is done on each client using server's X.509 certificate. These

validations confirm the identities of both the clients and the server. This allows SnortCM to prevent attacks which attempt to impersonate either the clients or the server components of SnortCM.

In addition to performing dual authentication and encrypting communication, the SnortCM server component uses an important security principle called "least privileges." Rather than binding to a TCP port below 1024 (which requires root access), the SnortCM server binds to port 3023. This allows SnortCM server to run at a lower level privilege using only the minimum access necessary to operate. This protects SnortCM from being "tricked" into performing a task detrimental to the operation of the system.

For the client and server communication, User Datagram Protocol (UDP) was considered instead of implementing Transfer Control Protocol over Internet Protocol (TCP/IP). However, TCP/IP was chosen over UDP because of its built in reliability. Furthermore, TCP/IP allowed SnortCM to utilize Secure Socket Layers (SSL). Finally, TCP/IP allocated a simple threading architecture for efficient communication with multiple (simultaneous) clients.

SnortCM interfaces with a MYSQL database since Snort™ is usually configured to log information to MYSQL databases. However, all database functions have wrappers written for them, meaning the team could make a few minor changes to allow the application to communicate with another database management system. The database is used not only for storing information, but also as a means of communication between different components. When a user issues a command from the User Interface, a record is inserted into a table in the database. On regular intervals, the server component acts as a scheduler and checks this table for commands. These commands are then sent over the network to the client component.

The client component translates the received commands into specific function calls. These function calls control the Snort™ process through various interprocess communication (IPC) mechanisms. It is important to note that the client component must reside on the same machine as the Snort™ process it is controlling.

The team elected to implement the client and server components in C++ because of the language's efficiency, portability and ease of use for the team members. The SnortCM User Interface is a web interface, designed in HTML and PHP. The web interface simplifies remote management, as users can access SnortCM from anywhere using a standard web browser. PHP is also designed to easily interface with a MYSQL database; again all database functions have wrappers written to allow for changes in the future. The user interface is accessed over a secure connection, HTTPS, which uses the Secure Socket Layer (SSL) protocol. When the user provides input in the form of text, the input is checked before it is used to prevent attacks such as cross-site scripting and SQL injection.

SnortCM Application

Shown in Figure 2 is a screenshot of the current version of SnortCM. After each user authenticates, they are transfer to the "Start Page." This is the main SnortCM page which presents users with status of their remote Snort™ sensors. As can be seen below, three sensors are being managed and monitored: "Snort Sensor1", "VPN" and "PS1". The "Snort Sensor1" status is red, meaning it requires immediate attention from the user. Furthermore, alert types are summarized in a simple bar graph, based on protocol and hit percentage: ICMP, UDP or TCP. This is key information for an administrator to assess the security of their network.

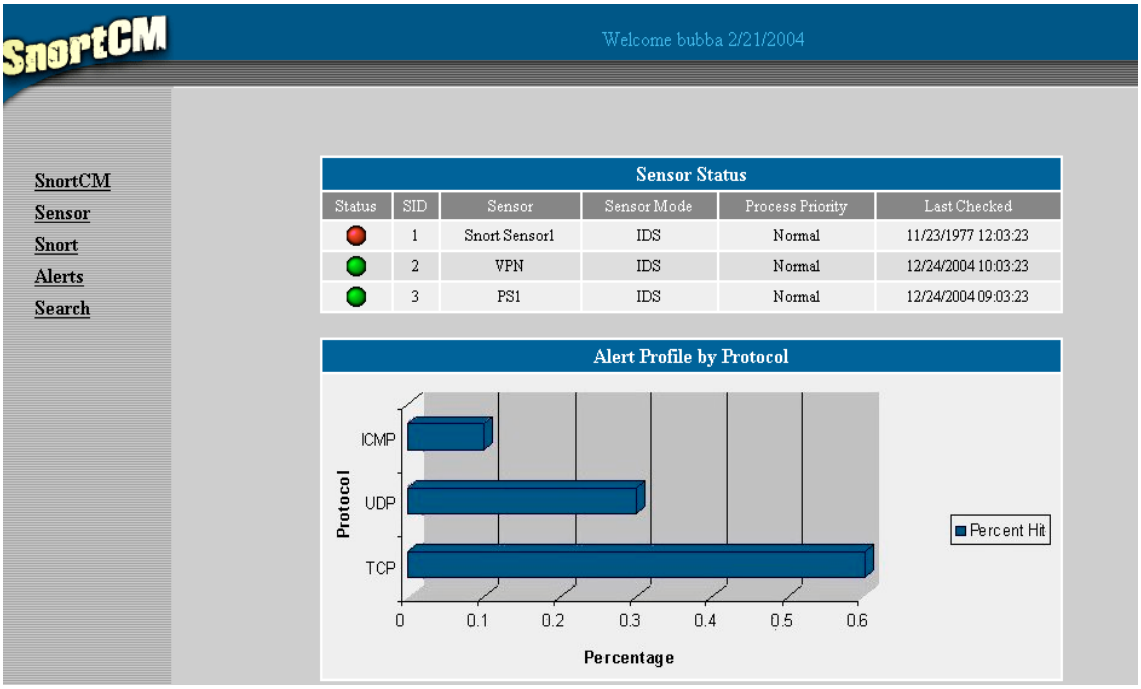

Figure 2: SnortCM "Start Page"

SnortCM is divided into five sub-menus: SnortCM, Sensor, Snort, Alerts, and Search.

SnortCM

This submenu gives the user the ability to manage SnortCM users and to set the SMTP server. Administrators can add a user to the SnortCM application and optionally set SnortCM to email the user alerts. This option will email the user if there are alerts related to Snort™ sensors. Additionally, a user can modify their password, email address, or enable/disable email alerts.

Sensor

The Sensor submenu gives users full control over all remote Snort™ sensors and their settings. On the main Sensor page, a table will display the current status of the sensors along with a graphical representation of the sensor network load. A user can *add* a sensor to SnortCM, by specifying the IP address, sensor name, log directory, configuration file, and optionally assign the sensor to a group (e.g. DNS, FTP, Telnet).

Given a user can add a sensor; a user can also *modify* a sensor. They can modify all the settings noted above, along with the option of removing a sensor permanently. Furthermore, users can *manage* each sensor. This is one of the strengths of SnortCM: being able to start, stop, restart or poll any given sensor, anywhere on the network.

Snort

The Snort submenu gives users the power to modify each sensor's Snort™ configuration settings. These options are organized for ease of use. A *Network* page gives the user the ability to specify things such as the sensor's IP addresse(s) for the network interface, ports etc. A *Preprocessor Plugins* page allows the user to include or exclude certain Snort™ plugins, based on the type of network they are protecting. A *Rules* page gives the user the flexibility to add a rules path, or add individual Snort™ rules and rule sets. An *Output Plugins* page lets the user include logging plugins to Snort™. Finally, the *Edit snort.conf* page, which allows the user (if they are knowledgeable in the inner-workings of Snort™) the ability to edit the raw snort.conf file.

Alerts

This option lists the last ten *Snort™ alerts*. Alerts are displayed in a table, giving the Sensor name, type of attack, source and destination IP addresses and the timestamp of the beginning of the malicious activity. A second option in the Alerts submenu displays the last ten *SnortCM alerts*, which would include any errors pertaining to the SnortCM application itself.

Search

This page gives users the option to search through history of attacks logged by the Snort™ sensors. The search options give maximum flexibility, allowing searching by (among other items): source/destination port, a given IP address range, along with a time of the attack. The results are neatly returned in a table for easy analysis.

SnortCM Advantages

Some systems exist that provide a GUI for viewing the information collected by Snort™ sensors. Other methods exist for managing multiple Snort™ sensors. Take for example the program Analysis Console for Intrusion Databases (ACID) [5]. ACID is an excellent program which can search and process data of security events generated by various firewalls, network monitoring tools, and intrusion detection systems; including Snort™. Although ACID is a great tool it doesn't have the capability to manage Snort™ systems. One such tool which can manage Snort™ sensors is called IDS Policy Manager [6]. IDS Policy Manager is a powerful way to modify Snort™ configuration and rules, but it lacks the ability monitor real-time attack alerts created by Snort™.

SnortCM provides all of this functionality in one package. SnortCM can manage multiple Snort™ sensors and both gather and report alert information. The benefit of being able to view the information from sensors and manage those sensors from one application should be apparent. When an attack occurs on a network, a quick response can prevent serious damage from occurring. Finally, the web-based User Interface of SnortCM allows users to assess and response to attacks from anywhere in the world.

Conclusion

As computer security threats continue to increase, a centralized intrusion detection system will be an invaluable tool for anyone managing a computer network. A system like SnortCM could allow fewer personnel to manage sensors and allow administrators to quickly identify and respond to possible attacks.

SnortCM gives IDS administrators an easier, more efficient method to administer IDS sensors deployed across multiple network segments. The system can also save corporations money by reducing the number of IDS administrators, as well as the amount of time required gathering logs from Snort™ sensors. By having all attack records stored in a single place, it allows the analyst much more flexibility in discovering attack patterns, and other attack issues which may have otherwise gone unnoticed.

References

1. Footpath Inc., (2002). Security Statistics. Retrieved Dec. 1, 2003, from http://www.footpath.com/articles/art00006.asp

2. CERT/CC, (2003). CERT/CC Statistics 1988-2003. Retrieved Mar. 1, 2003, from http://www.cert.org/stats/cert_stats.html

3. Raikow, David. (Oct. 24, 2001). *IDSs Bolster Network Defense*. Retrieved Mar. 4, 2004 from http://techupdate.zdnet.com/techupdate/stories/main/0,14179,2819361,00.html.

4. Roesch, Marty. (2004). Snort Web Site. Retrieved Mar. 4, 2004 from http://www.snort.org/

5. Analysis Console for Intrusion Databases (ACID), (2004). Retrieved Mar. 9, 2004 from http://www.andrew.cmu.edu/~rdanyliw/snort/snortacid.html

6. IDS Policy Manager, (2004). Retrieved Mar. 9, 2004 from http://www.activeworx.com/idspm/screenshots.htm