FREE AND OPEN SOURCE INTRUSION DETECTION SYSTEMS: A STUDY

SREENIVAS SREMATH TIRUMALA[1], HIRA SATHU[2], ABDOLHOSSEIN SARRAFZADEH[2]
[1] School of Computer and Mathematical Sciences, AUT University, Auckland, New Zealand
[2]Department of computing, UNITEC institute of Technology, Auckland, New Zealand
E-MAIL:ssremath@aut.ac.nz, hsathu@unitec.ac.nz, hsarrafzadeh@unitec.ac.nz

Abstract:

Importance of cyber security cannot be denied in the current cyber environment. With continuous growth of internet, cyber security has become a necessity for both big and reputed organizations as well as small businesses and individuals. Intrusion detection systems (IDS) are considered to be an efficient way for detecting and preventing cyber security threats. However, there has been not enough attention and awareness on intrusion detection and prevention systems, especially among small businesses and individuals. Due to this, selection and deployment of IDS is significance in regard to this subject being considered highly technical, expensive and time consuming process. To overcome this, it is necessary to create an awareness of IDS tools which forms the basis of this paper. This study is the first phase of an ongoing research. In this phase, we present a detailed study of three free and open source IDS tools which are most popular in their respective categories. The IDS software used for this study are Suricata, a Network based Intrusion Detection System (NIDS), Samhain, a Host Based Intrusion Detection System (HIDS) and Ironbee, a universal web application firewall system. This study of IDS tools at one place will serve as a knowledge source for both technical and non-technical audience, small businesses which may not afford experienced security consultants. Further, this will also help in identifying suitable IDS software for their respective organization.

Keywords:

intrusion detection system; cyber security; open source intrusion detection; Suricata; Samhain; Ironbee; IDS; HIDS; NIDS;

1. Introduction

Information security has been a pressing issue since the advent of IT. Rapid increase in development of internet and information systems had a direct impact on the increasing concerns about security. Technical technological advances on one side have also brought in information security challenges on the other side. With continually changing threat dynamics and strategies, defenders must keep pace with attackers. With continuous implementation of technological advances like having BYOD policies, security issues have grown exponentially [1].

The meaning and definition of cyber security is continuously evolving especially with respect to business enterprises and governments. Other aspects involved in defining cyber security are its applications and implementations. With respect to implementation and user domains, the definition and rules of adopting cyber security is distinct.

Table 1: Categories of Cybersecurity Threats [2]

Type	**Description**
Crime	Using systems for piracy, theft of information,& Financial fraud etc
Commercial Purpose	Organizations exposing sensitive information: personal records, product information, research plans etc. of competitors
Nation-state espionage	Involvement of government in acquiring sensitive government data from adversary agencies.
Warfare	Also known as cyber acts of war, involving governments, terrorist / extremist groups targeting a particular country or organization

For example, general users might be limited to their own password policy i.e., having a strong password, keep changing passwords etc., whereas for a network administrator, the password policy is just one of the aspects of security policy. Cyber risk is a term associated with various types of risks involved in being exposed to cyberspace or internet due to vulnerabilities. In other words when a device is exposed to cyberspace, possible threats involved constitute cyber risks [3]. When enterprises/organizations and individuals think of protecting their IT infrastructure they are actually addressing cyber risks. Securing IT environment (which includes data and infrastructure) from cyber risks is cyber security. With globalization, cyber based transactions have become part and parcel of an organization. This cannot be performed without exposing the organizational IT environment to cyberspace [4] and its ensuing risks.

The role of cyber security is to implement a secured environment for these internet based transactions. Securing an IT environment is considered as a better approach as the good old saying prevention is better than cure. Cyber security entails a strategy with understanding cyber risks due to vulnerabilities and proposing solutions for mitigating them. Cyber security vulnerabilities are security breaches or loopholes which expose the IT environment to attackers. Vulnerabilities will enable the attackers to compromise availability, confidentiality and integrity of the information [5]. Further, CVE defines vulnerability as software mistakes that can allow hackers to breach an IT environment and if need be take it over. Cyber threats [2], are categorized in the Table 2.

Threats are to be identified in order to apply the counter measures, and cyber security is not an exception. Forbes has identified a list of cyber threats with highest risk factors and their trends.

Table 2: Top Cyber Security Threats

Threat	**Risk Factor**
Social Engineering	Increasing
Advanced Persistent threats	Steady
Internal threats	Steady
Bring your own Device (BYOD)	Increasing
Cloud Security	Increasing
HTML5	Steady
Botnets	Increasing
Precious Target Malware	Steady

Standards are necessary for any implementations to ensure best practices. Cyber security standards are a set of globally accepted security standards to prevent cyber-attacks, by addressing cyber risks. Though there are some common ideologies, most of the time these standards are geographic, organization and / or country specific. However, there are certain international standards for analysis and assessment of cyber security policies (also known as ISMS family of standards) that are also the information security standards proposed by ISO and the IEC [6]. These standards are considered as a reference for analyzing and assessing security software. The standards that are published by ISO/IEC form the fundamental building blocks for preparing standards appropriate for the future technologies.

There is ongoing research on specific standards for intrusion detection and prevention systems. ISO/IEC 27039 ISO/IEC 27039 standards are for security techniques i.e., for selection, deployment and operation of Intrusion Detection and Prevention Systems (IDPS) [7] published in 2013. These forthcoming standards may change the entire scenario of intrusion detection software development and implementation. Some of the features that are expected to be incorporated in these standards are:

- Signature based Identification for common attack patterns
- Risks involved in configuration and deployment of IDPS
- Automating actions for common attacks and Incident response cycle
- Response reports with information of attacks and actions
- Blocking pattern less social engineering attacks, internal hackers and False alarms
- Additional Risks involving complex software

ISMS standards follow PACD (Plan-Act-Check-Do) model for implementation.

2 Intrusion Detection System (IDS)

As the name implies IDS detects/prevents intrusions/unauthorized entries into an IT environment. Its functionality can be compared with that of a burglar alarm. Typically IDS inspects all inbound and outbound activities and sends appropriate alerts to administrators or operators for further action. Sometimes IDS can be referred as an IPS when it covers active automatic prevention actions. Categorization of IDS is based on both the actions they take as well as on the systems they are ported on or the systems they are expected to cover.

IDS can be an Active IDS or a Passive IDS. Using an Active IDS, suspected attacks are automatically blocked, based on pre-programmed rules. This type of IDS is also referred to as Intrusion Detection and Prevention System - IDPS. IDPS offers real-time protection. Whereas Passive IDS only monitors the activities, logs the suspected activities and reports it to the administrator for action. Another type of classification is a Network based IDS (NIDS) and a Host based IDS (HIDS). NIDS collects information for network analysis based on wiretapping concept. Information to monitor includes traffic streams, network data. Network checks by NIDS are performed at random. NIDS are portable and monitors a particular network segment at a time. HIDS are detection systems pertaining to a single host. These systems monitor host activities. There can be a centralized control and activity logs for such monitoring. Most of the time HIDS monitor file systems. The source of HIDS is log files and system audit information. HIDS monitor integrity of the host as well as its communication with other computers. Sometimes a Web Application Firewall (WAF) is also considered as an IDS. WAF serves as a protection against cyber threats by monitoring HTTP traffic. WAF can be a software, server plug-in, a service or a daemon working in the background. WAF is required to protect the applications against threats which are considered as standard OWASP Top Ten Threats.

3 Assessment of standards for existing tools

3.1 Host based Intrusion Detection System - Samhain

Table 3: HIDS Tools

Tool	Free	Open Source	Full HIDS	custom modules	FIS Only	Mac support	Analysis on Host
Samhain	✓	✓	✓	✓		✓	✓
Tripwire		✓	✓	✓		✓	
OSSEC	✓	✓	✓	✓	✓		
Open DLP	✓	✓	✓		✓		
AFlick	✓	✓	✓		✓		
AIDE	✓	✓	✓		✓		

There are many HIDS available in the market with their own advantages and drawbacks. Some of the important features of various HIDS is presented as Table 3. Samhain is been chosen for the study because of its features and easy implementation. The nearest competition for Samhain is OSSEC. OSSEC performs the analysis at the server side whereas Samhain does it in client side which reduces the load on server, network traffic and enables more client CPU utilization.

Samhain is an open source host based intrusion detection system (HIDS) developed by Samhain Labs. Samhain can be deployed either centralized or on each of the computing nodes as individual implementation. Samhain HIDS provides the following functionality. Samhain has real time implementation running on over 200 servers. Some of the functions of Samhain are File Integrity Checking, File Monitoring and Analysis Root-kit Detection, Port Monitoring, Rogue SUID detection and Hidden process monitoring and analysis.

3.1.1 Features

Samhain supports multiple platforms like Linux, Mac OS, Solaris, AIX, and Windows with POSIX emulator. Further, the architecture of Samhain allows single host implementation as well as centralized client/ server implementation [8] as mentioned earlier. With this type of deployment the centralization of management is achieved. The Client/ Server implementation of Samhain is composed of components like Integrity Checker, Server component, client component, Yule server etc.

File/host integrity checker is a Samhain agent deployed on clients and servers. Server agent acts as a controlling agent whereas client agent acts as slave and is committed to the server. The client software works as daemon and is always at the service of the server. Yule Server is responsible for collecting reports, logs from the clients. The clients receive setups and updates through Yule. Yule maintains the statistics of various Clients and their attributes like Client-status etc.

Samhain has a database for maintaining the logs and reports. Relational databases like Oracle, MySQL and PostgreSQL are also supported. However, this is not mandatory. Beltane is the web based user interface for Samhain [9]. This is a separate package developed using PHP. Beltane is used to view reports, update client databases on server etc. Beltane-II is the latest commercial version. Deployment system is optional software that can be used to perform Samhain client deployments with ease.

Host integrity in Samhain is monitored using various modules. Every module is extensible which make Samhain a perfect choice. Samhain provides Windows Registry Check for windows based systems and Kernel Integrity check for Linux based systems. Further Samhain has separate modules for Open port monitoring, Log file monitoring and Analysis, SUID/SGID file check, Hidden Process check , Login/Logoff event monitoring and checking mounted devices.

File Integrity Check is for immediate notification of changes in the file systems. This will minimize the pressure on I/O operations and force or schedule checks. SELinux attribute check, file system attribute check (ext2 in case of Linux), POSIX ACLs. For Linux based systems, Samhain integrates with Kernel audit to find information about modified files like user, date and time etc. This operation is not possible in other OSs. This also performs checksum, size, owner, permission, file operations, users etc., level of sub directories can be set as a common parameter for all or as an individual value. Exclusion of files or directories is possible while performing checks. Supports User defined level checks. File checks can be scheduled to happen automatically. The server can request checks at any time by sending instructions to the Samhain client daemon

Samhain has a robust Log facilities with a central - server based log and a console - host based log. The logs are sent to the server t using Encrypted TCP connections. Every log file entry is signed to prevent unauthorized changes. Samhian has signature based email reporting. The log facilities of Samhain can be extended to execute addition logging software

Integrity has been a key aspect of Samhain. Samhain can be configured to hide its identity. This facilitates Samhain to be invisible to the intrusions. Connection to the server is password protected. This password is embedded into the executable. This password can be set only once. Samhain can always perform its operations in background without being noticed (Daemon Mode). But will still monitor and work normally, ceasing all unauthorized executable/ processes. Every executable built will have a unique 64-bit security key attached to it for extra protection to differentiate messages from intrusions / threats. Every message / report written to the log is signature based. Additional secured encryption is provided for the reports written on server.

3.2 Network based Intrusion Detection System - Suricata

Among the available NIDS systems, Suricata , Snort and Bro are commonly considered NIDS. Bro NIDS is UNIX / Linux based and does not support windows servers. Suricata is proven to be effective supported by already published comparative study [10]. Suricata was developed by Open Information Security Foundation (OISF). Some of the features of Suricata are detailed below.

3.2.1 Scalability

Most often firewalls and IDSs/IPSs are bottlenecks in the performance of network information systems. However, Suricata is configured to run each instance of the IDS process across multiple threads across different processors taking care of load balancing and performance. Thereby speeds in 10s of gigabits per second are realized which makes Suricata a highly scalable system.

3.2.2 Protocol Identification and support

It is common practice to filter traffic on port level basis. However, attackers bypass such filtering easily. After recognizing the common protocols running on the network, rules are written for the protocol and not for the expected port. This gives Suricata an exceptional capability of malware analysis and control. In addition, keywords can be matched within the protocol fields ranging from simple HTTP headers to a SSL certificate identifier. This IDS supports packet decoding for Layer 2 and Layer 3 and 4 protocols like, PPP, PPPoE, IPv4, IPv6, TCP, UDP, SCTP, ICMPv4, ICMPv6, GRE, SLL, VLAN, QINQ. Application layer decoding of: HTTP, SSL, TLS, SMB, SMB2, DCERPC, SMTP, FTP, SSH, and DNS is also supported.

3.2.3 File Identification, MD5 Checksums, and File Extraction

Multiple file types can be identified by Suricata that are transmitted over the network. Hence files can be both found and tagged for extraction using MD5 hashes. These hashes are calculated on the fly during extraction and writing of the files to disk. Further this IDS system can take the decision to keep these files or have them kept out of the system.

3.2.4 Performance

This IDS supports full multi-threading capability that speeds up network traffic analysis, providing much higher performance than established IDS like Snort system. Performance Statistics support use of a 64 bit machine to enable loading of multiple rule sets. It is observed from previous studies that Suricata gives least packet loss. Higher performance has also been confirmed where higher speed networks on free BSD was used [11].

3.2.5 Detection engine

Suricata rule set enables detection for capture, decode, classification of packets and even portions of packets. The detection covers both benign and malicious packets. The algorithms are based on multiple pattern matching that can be selected with a large range of configuration options. The rules can be reloaded with additional new rules on the fly, without having to restart Suricata. Studies have revealed that the detection results show least packet loss [11] and less false positives and less false negatives as compared to other NIDS. Alert generation as compared to snort was also 30% more for the dame rule set and same network traffic [10].

3.3 Ironbee Web Application Framework for Firewalls

Ironbee [12] is an open source Web Application Framework which can act as a skeleton for web application rewalls. The specialty of Ironbee is its exibility. When we have an existing Web Application Firewall that comes with the vendor specified set of features the users are bound to use the existing logic and framework for configuring the defence parameters in the firewall. Thereby, the defence system works according to the fundamental principles of the firewall. However, Ironbee, provides a high degree of flexibility to build the firewall according to the defence needs of the organization. Ironbee by itself can be implemented as a basic firewall without any modifications or additions or it may be customized. When compared to other Web application firewalls available in the market. Qualys the development organization termed Ironbee as a Universal Web Application Security Sensor for detecting Intrusions. Ironbee project was started in 2010 and became quite renowned in a short period of time. The fundamental principle of building Ironbee is contrastingly different to its contemporaries. Ironbee is developed on the concept of building a Universal Web Application Intrusion detection system. Ironbee [12] core engine has minimum functionality, just minimal processing of HTTP life cycle. The core module has an event subsystem and API interface which can be accessed by other modules. Data acquisition is done by an API plug-in deployed on the server which loads Ironbee engine and transfers the data. This makes Ironbee very flexible as it possesses the ability of being embedded in any security framework. It is also capable of passing HTTP data to the core engine. Ironbee works with almost all web servers and command line utilities. Ironbee implementation is based on modules which work in tandem with the core engine. These additional modules can be written in C, C++, Lua and can be extended to other programming languages.

Basic Ironbee implementation has the four core modules [12]. Configuration Library, Extension Modules, Ironbee Library and Server Components. Configuration Librart must be loaded with native server configuration and Ironbees own configuration. Extension Modules acts as additional modules to the fundamental core engine with loaded object libraries. Primary component consisting of core inspection engine are stored in Ironbee Library. Server Components are plug-ins or command line tools that drive the core Ironbee inspection engine.

3.3.1 Ironbee Rule management

Rules are configured by configuration file implanted by Ironbee rule engine. There are three types of rule matching approaches. Basic Matching will iterate through each data input and look for matches with specific operators. These rules are limited to be executed only once in a cycle. In Stream Matching there will be data buffering, with which the stream of data is analyzed, in small pieces rather than analyzing a continuously large data stream. This methodology ensures that there is no requirement of a large buffer. This will further allow inspection of limited fields required, rather than all the fields. External Rules are written and implemented to build customized comparison. These rules that are defined and configured externally can be more expressive and flexible. While inspecting and analyzing, Ironbee generates some events with eventId, event type, observation, data, fields, messages, severity etc. This information is helpful for analysis and status reporting. Request and Response Headers handling. Usually, Request and Response headers are small in size and are buffered for inspection. On the other hand the Request and Response data bodies will be very large and is in turn the main target of attackers. This particular problem is addressed by Ironbee using four procedures, Inspection, Processing, Buffering and Logging.

In Inspection, Response and Request bodies are inspected closely. With Processing, multiple types of analysis is carried out on how Request / Response body is processed?. This is followed by Buffering in which entire Request/Response block will be buffered for processing for efficient detection. Though it is possible to inspect as we go, this may cause actions that may be dubious in nature. The entire life cycle of inspection is recorded by Logging procedure.

Ironbees built-in processing engine has a default processing logic for Request / Response data body processing. There are many advantages by proposing Ironbee with a universal standard. Custom format for processing can be added to enhance its processing capability. Ironbee inspects data which is in two forms - fields or streams. The HTTP data can be of fields or collection of fields. These fields can be custom configured.

3.3.2 Application Features

Generally Software selection is based on performance, rating, productivity and price. Due to this, organizations tending to implement different products with variable configurations and settings. There will however, be some collaborative similarities, with different configurations and responses. Hence, developing a standard framework across different software products will be quite proficient and beneficial in selecting the right product.

Ironbee is designed in such a way that its compatibility will enable its reuse of the code / rules at different places. When an application is deployed on various platforms a standard rule can be made applicable irrespective of platforms. Ironbee has multiple deployment modes to preserve its universal deployment capabilities which are Passive Mode, Embedded Mode, Reverse Proxy, Command line processing. The portable version of Ironbee requires a tiny interface layer which is used to acquire data for processing. This will allow Ironbee deployment in an independent environment. Environment dependent user interface will give comfort for different users working in different environments. This includes easy to configure, auto-configuration, and a separate advanced configuration for advanced users.

3.3.3 Functional Features

- User based interface with multiple deployment modes
- Short and long term activity tracking with historic data
- Real time entities based data model
- Multiple and mixed pattern matching capability
- Policy based decisions
- Interoperability with other applications and security systems with data exchange capability

Ironbee can be configured for any of the Seven security models available i.e. DOS and DDOS attack detection, Pattern monitoring Secure XML Parsing and XML Validation, Security policy for Contents, Brute force attack recognition, Vulnerability scanning (both active and passive) and Cookie encryption and digital signing.

Ironbee has inbuilt traffic monitoring and analysis for both inbound and outbound traffic for known exploits and vulnerabilities in most used applications. Ironbee covers both application and protocol level evasion techniques. Ironbee has blacklisting and white listing based on package patterns for most common attacks. Ironbee supports 100% implementation of custom logic which can be divided upon 80%, 19% and 1% for rule based implementation, scripting platform and compiled module support for high performance.

4 Conclusion and Future Direction

In this paper, we attempt to emphasize the importance of intrusion detection systems (IDS) and provide an analysis of three significant open source tools: Samhain, Suricata and Ironbee. We also presented, briefly, various types of cyber security threats, their trends and various ISO standards along with an insight into various IDS tools. This paper further provides technical and functional analysis of IDS tools to give better understanding for the individuals and small businesses that are not well versed with these technologies. This study will serve as a reference in selecting suitable IDS tools depending on requirements based on purpose, risk and features. This paper gives a theoretical understanding of IDS tools and the future extension is anticipated to cover an experimental analysis of these tools when deployed in a small to medium organization, along with the implementation frame work which would further assist in selecting suitable IDS tool.

The next phase involves an empirical study of open source and proprietary IDS tools, with a view to establish the effectiveness of each. This phase of study would be followed by researching the effectiveness of IDS tools in different cloud environments.

References

[1] W. Zhao and G. White, “A collaborative information sharing framework for community cyber security,” in *Homeland Security (HST),IEEE Conference on Technologies for*, pp. 457–462, IEEE, 2012.

[2] D. Kriz, “Cybersecurity principles for industry and government: A useful framework for efforts globally to improve cybersecurity,” in *Cybersecurity Summit (WCS), 2011 Second Worldwide*, pp. 1–3, IEEE, 2011.

[3] B. Al Sabbagh and S. Kowalski, “St (cs) 2-featuring socio-technical cyber security warning systems,” in *Cyber Security, Cyber Warfare and Digital Forensic (CyberSec), International Conference on*, pp. 312–316, IEEE, 2012.

[4] G. Jakobson, “Mission-centricity in cyber security:architecting cyber attack resilient missions,” in *Cyber Conflict (CyCon), 5th International Conference on*, pp. 1–18, 2013.

[5] W. W. Agresti, “The four forces shaping cybersecurity,” *Computer*, vol. 43, no. 2, pp. 0101–104, 2010.

[6] I. . IEC, “Iso/iec 27001:2013 standards,” 2013.

[7] I. . IEC, “Iso/iec 27039 standards,” 2014.

[8] samhainLabs, “Setting up a client/server samhain system,” 2014.

[9] samhainLabs, “Beltane,” 2014.

[10] E. Albin and N. Rowe, “A realistic experimental comparison of the suricata and snort intrusion-detection systems,” in *Advanced Information Networking and Applications Workshops,26th International Conference on*, 2012.

[11] K. Thongkanchorn, S. Ngamsuriyaroj, and V. Visoottiviseth, “Evaluation studies of three intrusion detection systems under various attacks and rule sets,” in *IEEE TENCON*, 2013.

[12] B. Rectanus, “Ironbee reference manual,” 2014.