# Intrusion Detection using Open Source Tools

Jack TIMOFTE
jack.timofte@gmail.com

*We have witnessed in the recent years that open source tools have gained popularity among all types of users, from individuals or small businesses to large organizations and enterprises. In this paper we will present three open source IDS tools: OSSEC, Prelude and SNORT.*

***Keywords**: Network security, IDS, IPS, intrusion detection, intrusion prevention, open source.*

## Introduction

The traditional form of securing the network, the firewall proved to be insufficient. Given the nature and the complexity of the attacks, new ways of protecting the network had to be developed. Intrusion Detection Systems, or shortly IDS are tools to monitor the events occurring in a computer system or network and detect signs of possible incidents, such as violation of computer security policies, acceptable use policies or standard security practices. An Intrusion Prevention System, or IPS in addition to detecting incidents, can play an active role by attempting to stop the possible incidents which are detected. The Intrusion Detection/Prevention Systems (IDPS) can be classified according to different criteria. The NIST Guide to Intrusion Detection and Prevention Systems (NIST 800-94), lists four main types:

- Network-Based, which monitor the network;
- Wireless - monitor wireless network traffic;
- Network Behavior Analysis (NBA) - which examine the network traffic to identify threats like denial of service attacks and malware;
- Host-Based - monitor a single host and the events occurring within that host.

According to the way an IDPS detect the incidents, they can be classified into three categories: signature-based, anomaly-based and stateful protocol analysis, but most IDPS systems use multiple detection methodologies, either separately or integrated, to improve the performance.

Below we will briefly present a selection of three open source tools: OSSEC, Prelude and SNORT.

**OSSEC** (www.ossec.net) is an open source host-based intrusion detection system (HIDS).

According to their own website, *"OSSEC is a scalable, multi-platform, open source Host-based Intrusion Detection System (HIDS). It has a powerful correlation and analysis engine, integrating log analysis, file integrity checking, Windows registry monitoring, centralized policy enforcement, rootkit detection, real-time alerting and active response. It runs on most operating systems, including Linux, OpenBSD, FreeBSD, MacOS, Solaris and Windows."*

In the recent years, OSSEC has received several awards. In 2007, it was placed on the first position in the *Top 5 Open Security Tools* in the enterprise by LinuxWorld. In 2006, OSSEC was voted as the second best IDS tool by the survey conducted by the **sectools.org** website. Recently, OSSEC has been acquired by Third Brigade, but according to the press release, OSSEC will remain open source.

The OSSEC HIDS can be installed as a stand-alone tool to monitor one host or can be deployed in a multi-host scenario, one installation being the server and the others as agents. The server and agents communicate securely using encryption.

OSSEC also has intrusion prevention features, being able to react to specific events or set of events by using commands and active responses. The system allows the creation of new commands which can be bound to events. The system comes with some predefined active response tools, but the administrator can add others.

OSSEC was designed initially for Linux, but

it evolved and since version 0.8 it also features a Windows agent, which can monitor the event log and other files.

**PRELUDE**. Prelude (www.**prelude-ids**.com) is more than an open source IDS system - it is a framework which enables other security applications to report to a centralized system.

It makes use of the IDMEF (*Intrusion Detection Message Exchange Format*) standard proposed by IETF, which allows defining the events recorded by different sensors using a single language. Prelude is considered a hybrid system, since it allows the coexistence of the event data from host-based, network-based or wireless IDS agents, or simply any other security application. Existing security applications can be modified to use the Prelude system, using the provided C, Python and Perl frameworks.

The main components of a Prelude system are: the *Prelude library* (framework), the *Prelude Manager*, the *Prelude-LML*, *PreludeDB* library and *Prewikka* (the console). The Prelude library, Libprelude, is used to access the Prelude system and provides an API (Application Programming Interface) to create events in the IDMEF (*Intrusion Detection Message Exchange Format*) format. It can be used for failover and allows the creation of sensors that read the events received by one or a set of prelude managers. The Prelude Manager is a high-availability server which collects and normalizes information from distributed sensors and stores them into the database, having several features such as relaying or filtering the events.

The Prelude LML is a signature-based log analyzer monitoring log files and received syslog messages for suspicious activity. It can handle events generated by a large set of components, such as: Cisco PIX, Clamav, ipchains, Netfilter, ipfw, Nokia ipso, MsSQL, Nagios, NTsyslog, Pam, Portsentry, Postfix, Proftpd, ssh etc.

The PreludeDB library allows the developers to use the Prelude IDMEF database, providing an abstraction layer based upon the type and the format of the database used to store IDMEF alerts.

The last component, Prewikka is a console providing advanced features like contextual filtering, aggregation, etc. Below we can see a simplified architecture with three sensors.

An interesting feature of Prelude is the possibility to “relay” the events between managers: in the following example, Branch A is relaying all the events to the Prelude Manager located in the Network Operation Center of the organization. Using this setup, Branch A can only access the events recorded by sensors D, E and F, while the NOC (Network Operation Center) can access the events generated from all branches.

An interesting thing related to the three open source tools presented in this article, is that they can be integrated: Prelude IDS framework has native support for both Snort and OSSEC. A list with all the external sensors which are supported natively is presented in Table 1.

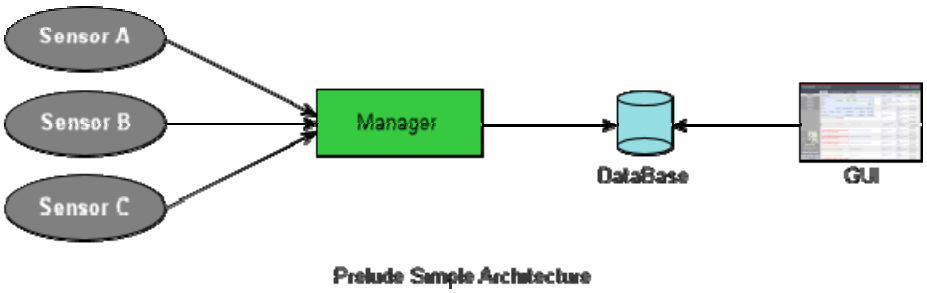


**Fig.1.** Prelude Simple Architecture

Using the IDMEF API, additional own sensors can be defined and programmed. In the PRELUDE documentation there is such an example written in C.

**SNORT**. Snort (www.**snort**.org) is, without doubt, one of the most popular open source security tools. With millions of downloads, it is used by individuals as well as large corpo-

rations or government organizations. The first version was written in 1998 by Martin Roesch, who later founded Sourcefire. Since then, the product evolved both as features and as portability: currently Snort is available for most major platforms including Windows, BSD, Solaris or Mac OS X. It is worth mentioning that Snort has an excellent support from the user community. This can be considered as a big advantage since the availability of signatures for new attacks can be faster than for most commercial IDS tools. According to Jennifer Albornoz Mulligan, security researcher at Forrester Research, "*Once the community writes those signatures, Sourcefire takes a little extra time to test them before it puts them out there, but the process is still generally more responsive than others*". This community ruleset comes in addition to the official ruleset, released by the Sourcefire Vulnerability Research Team.

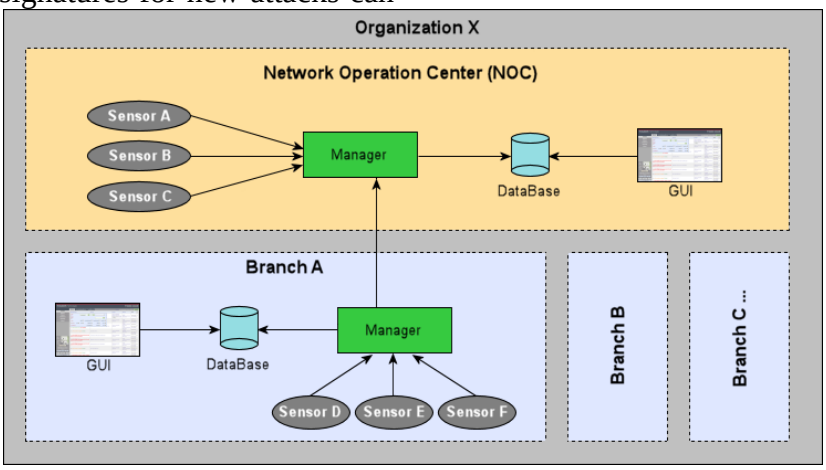


**Fig.2.** Prelude Relaying Architecture

| Security Tool | Description |
|---|---|
| **AuditD** | audit system |
| **Nepenthes** | detection and collection of worms and malware |
| **NuFW** | authenticating firewall |
| **OSSEC** | HIDS |
| **PAM** | authentication tasks |
| **PFLogger** | reports alerts from OpenBSD firewall |
| **Sancp** | information collection on network activity |
| **Samhain** | file integrity checker |
| **Snort** | NIDS |

**Table 1.** Native Support in Prelude IDS

Snort can run in different modes, ranging from a simple sniffer to a IPS system:
- Sniffer mode;
- Packet Logger mode - logs packets to disk;
- NIDS mode - allows Snort to analyze network traffic for matches against a user-defined rule set and to perform several actions based on these matches;
- Inline (IPS) mode - allows Snort to drop or pass packets based on the specific Snort rules.

Working as an IDS, Snort uses preprocessors and rules.

*Snort Preprocessors* allow the functionality of Snort to be extended by allowing users and programmers add modular plug-ins. Preprocessor code is run before the detection engine is called, but after the packet has been decoded. Such preprocessors exist for IP defragmentation (Frag3), TCP stream reassembly (Stream4), HTTP, FTP, SMTP, SSH etc.

*Snort rules* are used by the system to detect incidents. Snort rules are divided into two logical sections, the rule header and the rule

options. The rule header contains the rule's action, protocol, source and destination IP addresses and netmasks, and the source and destination ports information. The rule option section contains alert messages and information on which parts of the packet should be inspected to determine if the rule action should be taken.

At this time, the latest stable version is 2.8.2.2. Currently a new beta version is under testing, Snort Security Platform (SnortSP) v 3.0, which comes with a new architecture (see Figure 3 below). The primary components are the **Snort Security Platform (SnortSP)** and **traffic analysis engine modules** that plug into SnortSP. The current beta version includes one engine module containing the Snort 2.8.2 detection engine. Its major features are:

- Shell-based user interface with embedded scripting language;
- Native IPv6, MPLS and GRE support;
- Native support for inline operation;
- More subsystem plugin types such as data acquisition modules, decoders and traffic analyzers;
- Multithreaded execution model - multiple analysis engines may operate simultaneously on the same traffic;
- Better performance.

The main components of the new architecture will be briefly presented below.

The *Data Source* component encapsulates the common functionality needed by any network traffic before the analysis tasks and incorporates several modules:

- *The Data Acquisition Module (DAQ)* – gets the packets from the underlying hardware – Snort 3.0 can incorporate arbitrary packet interfaces like libpcap, IPQ etc;
- *The Decoder* – which performs the same tasks like in Snort 2.x: validate the packets, detect protocol anomalies and provide a referential structure;
- *The Flow Manager* – can help tracking the conversations on the network;

*The IP Defragmenter* – can provide IPv4 and IPv6 services for putting the packets back together and for fragment reassembly.

The *Action System* handles event queuing, notification and logging when the system fires events. The supported types are text (console), syslog and Unified 2 (a serialized binary stream format).

- *The TCP Stream Reassembler* – provide target-based services for reassembling TCP segments into normalized streams.

The *Data Source API* is an interface between the *Data Source* component and the *Dispatcher*.

The *Dispatcher* coordinates the information flow between the different components of Snort 3.0.

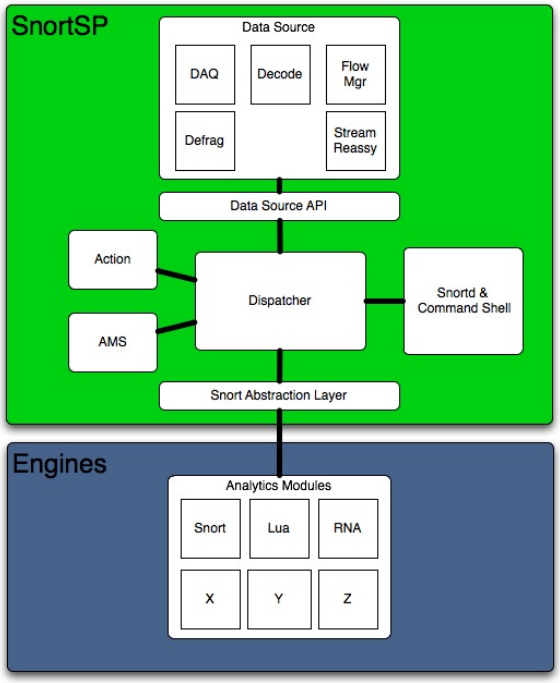


**Fig.3.** Snort 3 Architecture

The *Attribute Management System (AMS)* is used to store network data about the operational environment in which the Snort system is deployed.

The *Analytics System* contains the detection engines. The idea is to have the detection in analytics modules that run as separate threads. Using this architecture, several threads can operate simultaneously on the data coming from the dispatcher. Currently among the analytics modules under development are the Snort 2.x engine, RNA (for Sourcefire implementation) and a Lua (a scripting language) traffic analysis module.

*Snortd and command shell.* Snortd is the daemon process, and the command shell pro-

vides management services for different software modules, processes and threads, health management and includes the Lua scripting language.
While Snort does not offer a GUI, there are many complementary open-source tools like ACID (PHP-based) or BASE (Basic Analysis and Security Engine) which provide this GUI functionality for Snort. Or it can be integrated into an IDS framework, like Prelude IDS. There are also other tools, a simple query on freshmeat.net resulting in a listing of 68 projects, at the time of the writing of this article. Of course, we do not bring into discussion the quality of all these projects, not all of them being developed to a 'mature' level.
Sourcefire, the company which maintains SNORT, offers also commercial solutions based on the Snort Engine – Sourcefire IPS and Sourcefire RNA . It is worth to be noted that in 2006, the Israel based company Checkpoint intended to acquire Sourcefire, the owner of Snort, but, according to the media [7], "FBI and Pentagon expressed strong reservations about the deal because Snort is used to safeguard classified U.S. military and intelligence data." As a consequence, the deal was off, but we can deduct the importance given to Snort.

## Conclusion

The three tools presented above are open source, but commercial support is available from the companies maintaining the products. The fact that commercial companies are behind these projects, while the products remain open source, can be seen as a good sign, allowing the tools to be further developed in an organized manner and providing the necessary funding. The intended users for the tools listed here are not only individuals, but also businesses and other organizations. Getting a list with the companies using OSSEC, Prelude or Snort is not an easy task, since most organizations prefer not to expose their security architecture. For example, OSSEC lists only two companies from Brazil: Embratel and Resenet ISP. The Prelude IDS homepage has different sections by industry type, like Financial, Manufacturing, Public etc, only briefly presenting one real case study per section, but without naming the respective organizations.
However a question arises: are open source IDS products good enough for corporate use? According to a study from Gartner [9], some commercial IDS providers use Snort and/or Snort signatures for their appliances, which gives Snort credibility in this direction. But different organization have different needs. While for big enterprises open source IDS systems might not be enough, since they are the target of the most sophisticated attacks, the smaller and medium-sized businesses can protect their intranet for free or at a fraction of the price of a commercial system. However, even for the companies choosing commercial IDS, tools like Snort, OSSEC or Prelude can be used for 'a second opinion'.

## References

1. Karen Scarfone, Peter Mell, *NIST 800-94, Guide to Intrusion Detection and Prevention Systems (IDPS)*, Feb 2007
2. Open Source Intrusion Detection: No-cost System Lockdown – http://itmanagement.earthweb.com/secu/article.php/11076_3673721_1
3. OSSEC Homepage - www.ossec.net
4. SNORT Homepage - www.snort.org
5. Prelude IDS Homepage - www.prelude-ids.com
6. The Intrusion Detection Message Exchange Format (IDMEF) http://tools.ietf.org/rfc/rfc4765.txt
7. The Case for Open Source IDS, http://www.itsecurity.com/features/the-case-for-open-source-ids-022607/
8. Check Point drops plans to acquire Sourcefire - http://searchsecurity.techtarget.com/news/article/0,289142,sid14_gci1175561,00.html
9. Magic Quadrant for Network Intrusion Prevention System Appliances, 1H08 - http://mediaproducts.gartner.com/reprints/tippingpoint/154849.html
10. Snort 3.0 Architecture Series Part 1: Overview, http://securitysauce.blogspot.com
11. Snort 3.0 Architecture Series Part 2: Changes and Betas, http://securitysauce.blogspot.com