SOURCEfire network SECURITY

SNORT™ 2.0
Detection Revisited

Sourcefire, Inc.
9212 Berger Road
Suite 200
Columbia, MD 21046

April 2004

www.sourcefire.com

TABLE OF CONTENTS

ABSTRACT

Sourcefire's commitment to delivering the most innovative and effective intrusion management solutions continues with the latest contribution to Snort 2.0 development. As part of Sourcefire's dedication to the Open Source community, the company continually upgrades Snort with technologies and enhancements developed for its commercial products. These enhancements provide users with increased accuracy and up to 18 times greaterperformance than previous versions.

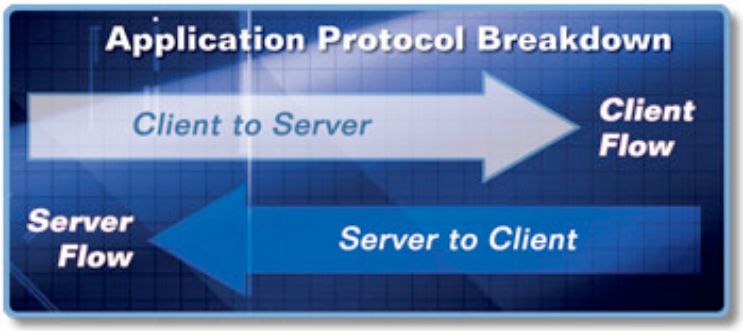
Snort 2.0 has been reengineered to use a new HTTP Protocol Flow Analyzer and Detection Engine. The Flow Analyzer optimizes data flow by reducing unnecessary data inspections while the Detection Engine uses a fast setbased rule selection methodology and a high performance multi-pattern search engine. The multi-pattern search engine uses a two-stage architecture to inspect data and find rule matches. The first stage of the multi-pattern search engine is a high-speed set-based inspection engine, which quickly identifies potential rule matches based on content and ports. The second stage is an enhanced rule processing engine, which provides additional functionality for in-depth validation of potential rule matches. Together, these enhancements greatly improve the performance and efficiency of Snort and help to reduce false alarms.

PROTOCOL FLOW ANALYZER

The Protocol Flow Analyzer classifies network application protocols into client and server data flows. In-depth analysis of these protocol data flows allows Snort to make intelligent decisions about protocol inspection.

What is a Protocol Flow?

A protocol flow refers to the client or server communication in an application protocol. For example, HTTP client-to-server communication is considered a flow and HTTP server-to-client communication is considered a separate flow. This allows Snort to break down a particular application protocol into two distinct flows, a flow destined for the client and a flow destined for the server.



Protocol Flow Analysis

Protocol flow analysis is performed at a high level and is usually only concerned with a few important aspects of a particular protocol flow, such as a server response code or a client request type. Flow analysis does not replace other protocol inspection technologies; instead it complements them. It is a generic analysis that allows an application protocol to be classified as a client or server flow. Once an application protocol is classified into a client flow and a server flow, it gives Snort useful knowledge as to the type of inspection and the regions of the protocol flow to inspect.

Benefits of Protocol Flow Analysis

Protocol flow analysis gives Snort rudimentary knowledge of a particular application protocol. With this knowledge Snort can determine which, if any, part of a protocol to inspect. This significantly reduces processing time and reduces false positives by limiting the amount of inspection that is done.

DETECTION ENGINE

The New Detection Engine is broken into three distinct technologies or phases. The Rule Optimizer, the Multi-Rule Search Engine (this includes the standard Snort validation), and the Event Selector.

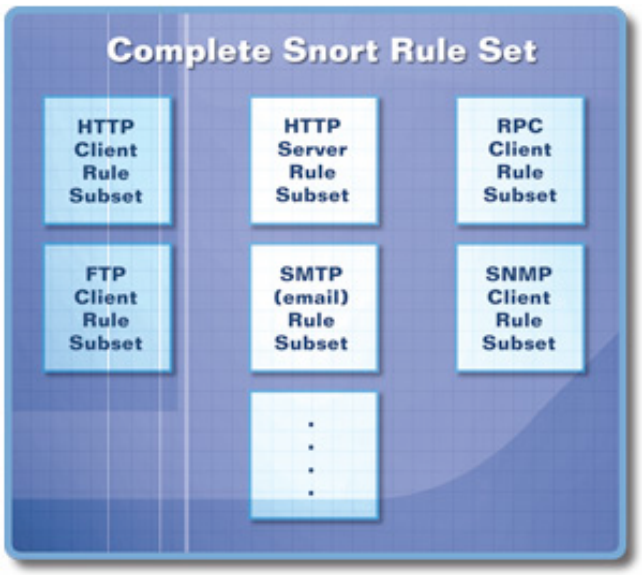
Rule Optimizer

The Rule Optimizer utilizes a set-based methodology for managing Snort rules and applying them to network traffic. Rule subsets are formed based on unique rule and packet parameters using a classification scheme based on set criteria. This allows the entire Snort rule set to be divided into smaller subsets of rules based on these unique parameters.

The benefit of this set-based methodology is that the Snort rule subsets are predetermined during initialization. Since these subsets are based on the unique rule parameters such as Source port, Destination port, and Rule Contents, each rule subset consists of the complete set of rules that are applicable to each packet. This guarantees that all applicable rules are tested against each packet, and ensures rules that cannot possibly match the packet are ignored.

Rule Processing Initialization

When Snort begins running, it reads and parses all the activated rules. The Snort rules are then passed to the Rule Classifier, which classifies them into subsets. This is done prior to any packet or stream processing. Once the Snort rules have been divided into subsets, each incoming packet is matched to a corresponding rule set based on the packet's unique parameters. For example, if Snort is run with 1500 rules, these 1500 rules get divided into smaller subsets based on transport and application-layer protocols. So 500 of these rules may go into the HTTP client rule set and another 50 rules could go into HTTP server rule set, etc.



Rule Set Selection

Once Snort proceeds to the rule processing stage for each packet, the packet parameters are passed to the Rule Manager to select the appropriate subset of rules to apply to a packet. Once the rule set is selected, the multi-rule search processing begins.

Multi-Rule Search Engine

The Multi-Rule Search Engine is broken into three distinct searches based on unique Snort rule properties:

1. **Protocol field search**
   The protocol field search allows a rule to specify a particular field in a protocol to search. For example, Snort uses the 'uricontent' keyword to search HTTP request-uri fields.
2. **Generic content search**
   The generic content search allows a rule to specify a generic byte set to match against the payload. For example, this functionality is used to look for buffer overflows in all packet payloads and can also be used by Snort users to search for any ASCII or binary byte sets that may signify an attack on their network.
3. **Packet anomaly search**
   The packet anomaly search allows a rule to specify characteristics of a packet or packet header that is cause for alarm. Packet anomaly rules do not have any type of content searches and are focused on the packet's other characteristics. While the three search types can utilize anomaly detection, the packet anomaly search is a specific type of detection. An example of a packet anomaly rule is one that looks for an ICMP packet with over 800 bytes.

Rule Inspection

The search engine uses a configurable high performance multi-pattern search engine to find all occurrences of protocol field and generic content patterns. The packet anomaly rules are processed using a search scheme based on the standard Snort rule processing. When a match is found in any of these three search types, the standard Snort processing fully validates the specific Snort rule. If the Snort rule is validated, an event is generated and added to the event queue. Once the search engine has completed processing the packet, the Event Selector processes the event queue.

Event Selector

Event Selection

The event queue allows Snort to track every occurrence of every rule match event within a packet. The event selector then prioritizes events from the event queue and selects events based on the assigned priority. Currently, the event with the longest single match is considered to have the highest priority and is selected. This event is then sent to the Snort output system.

ENHANCED RULES LANGUAGE

Distance and Within

The Snort rules language has been extended to support describing the number of bytes between content matches within a packet, in effect allowing stateful pattern matching. This allows the Snort rule language to describe anomalies in application layer protocols as well as imparting the concept of pattern ordering. The pattern ordering syntax allows Snort to represent the simple type of regular expression "***pattern1.*pattern2***".

What this provides effectively is the ability to model application layer protocols with Snort's content matching system, from RPC to HTTP to IMAP. This is an extremely useful capability, for example it allows Snort to have signatures that alert if a particular number of bytes have been seen following a command character without finding a line termination, allowing new buffer overflows to be found even without having a specific signature for them.

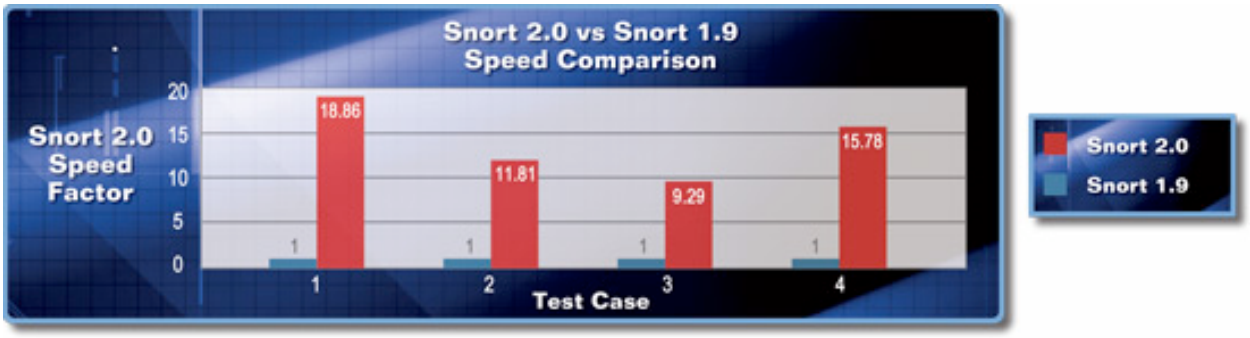
TCP State Enhancements

The rule language has been improved to isolate the server side and client side of a TCP conversation. This helps reduce false positives tremendously as odd high port signatures that were previously set off by talking to http servers may now have a new constraint added. This allows traffic from port 80 to be checked to see if it is client side or server side traffic.

In order to bring this enhancement in with the already existing state tracking recovery, if the state of a TCP stream is in doubt Snort will act on the side of caution and treat the packet both as a client side and server side traffic. This is another way that Snort will defend itself in the midst of IDS based attacks.

PERFORMANCE ENHANCEMENTS

The combination of optimized data flow, enhanced rule selection and a new high performance multi-pattern search engine, gives Snort 2.0 up to eighteen times the processing speed of Snort 1.9.



Sourcefire, Inc.

Sourcefire leverages the industry's most widely deployed and respected IDS as its core detection engine and has assumed the industry responsibility of maintaining and upgrading the Snort IDS. In a recent online poll conducted by the Security Administrator newsletter, 92 percent of respondents use Snort to conduct IDS on their networks1. Snort utilizes a combination of protocol analysis, anomaly detection, heuristic analysis, and rule based inspection techniques to perform the most thorough Intrusion Detection analysis available anywhere, and at wire speeds even beyond Gigabit.

Sourcefire, Inc. is a network security company protecting enterprises and government against the threat of network attacks and misuse. The company was founded in 2001 by the original creators of the open-source Snort Intrusion Detection System (IDS), the most widely deployed IDS, that forms the foundation for the Sourcefire product suite. Today, Sourcefire combines the Snort technology with sophisticated proprietary technologies, professional data analysis and management tools, along with best practices from respected security industry experts. Sourcefire is a privately held company headquartered in Columbia, MD. For more information about Sourcefire, please visit www.sourcefire.com.

[1] Results of poll conducted on Security Administrator site October 8, 2002 http://www.secadministrator.com/Index.cfm#Poll