



WHITE PAPER

SNORT[®] THREAT PREVENTION COMPONENTS



KNOW MORE NETWORK RISKS
NO MORE GUESSING

INTRODUCTION

Snort's threat detection and prevention components work together to reassemble traffic, prevent evasions, detect threats, and output information about these threats without creating false positives or missing legitimate threats.

The threat prevention process in Snort consists of multiple components which work together to reassemble traffic as a target host would see it, identify traffic areas that may contain threats, and match Snort rules against these traffic areas to recognize attacks. Together, these components efficiently detect threats and reduce or eliminate false alarms.

The threat prevention components of Snort include a packet classifier, which decides which packets are inspected; an IP defragmenter and a TCP reassembler, which ensure Snort inspects IP fragments and TCP segments in the proper order; a portscan processor, which watches for portscans, and a detection engine, which performs protocol normalization, rule matching, and many other detection functions.

Snort's detection components reside at the core of the threat prevention capabilities of Sourcefire Intrusion Sensors. They ensure that threats are detected, false positives and negatives are avoided, and detection performance is high.

BENEFITS

The threat prevention components of Snort:

- Select traffic to be inspected, ensuring that only traffic that could be vulnerable is inspected. Looking at only relevant traffic helps Snort's performance and limits the potential for false positives.
- Normalize traffic to ensure that it is in a consistent format and in the proper order prior to inspection. By putting traffic in a consistent format and in the order that the target sees it, Snort avoids the potential for evasion.
- Match traffic against one of the industry's largest rulesets using a number of different detection methods. By using multiple detection methods and using a large ruleset, Snort ensures that a wide variety of threats is detected and false negatives are avoided.

THREAT PREVENTION COMPONENTS

PACKET CLASSIFIER

Snort begins processing the traffic it sees by classifying it at a high level. The classifier decides whether a packet is:

- Token Ring/Ethernet/etc.
- VLAN/Not VLAN
- IP/Not IP
- TCP/UDP/ICMP

IP DEFRAGMENTER

Once packet classification is complete, Snort defragments IP packets.

IP fragmentation is the splitting of an IP datagram into two or more smaller IP datagrams. Typically IP datagrams are fragmented when the maximum size of a frame (MTU) is exceeded, although they can be fragmented for other reasons. IP fragments can arrive in any order at the destination system, and can reach the destination using different paths. The destination system buffers IP fragments until all the fragments of a datagram are received, following which it processes the datagram.

Effective IPS implementations must handle:

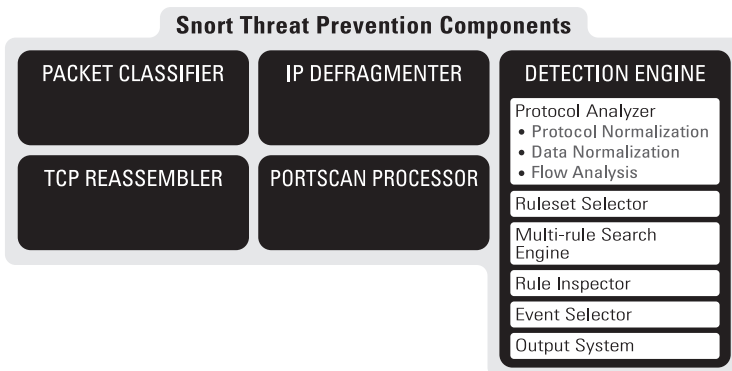
- IP fragments arriving out of order
- IP datagrams that never complete
- Incorrect IP options

With certain IPS implementations, attackers can take advantage of IP fragmentation by exploiting the fact that different operating systems and IP implementations reassemble IP fragments differently under unusual circumstances. These circumstances can include:

- Overlapping IP fragments
- Retransmitted fragments

Snort has a module called frag3 that serves as its IP defragmenter. In order to detect attacks effectively, frag3 must reassemble IP datagrams in the same way that the destination system for the attack would. Target-based analysis is a relatively new concept in network-based intrusion prevention. The idea of a target-based system is to model the actual targets on the network instead of merely modeling the protocols and looking for attacks within them.

In an environment where the attacker can determine what style of IP defragmentation is being used on a particular target, the attacker can try to fragment packets such that the target will put them back together in a specific manner while any passive systems trying to model the host traffic have to guess which way the target OS is going to handle the overlaps and retransmits. If the attacker has more information about the targets on a network than the IPS does, it is possible to evade the IPS.



This is where the idea for “target-based” or “adaptive” IPS comes from. For more detail on this issue and how it affects IPSes, read the Ptacek & Newsham paper at <http://www.snort.org/docs/idspaper/>

The basic idea behind target-based IPS is that the IPS is told information about the target hosts on a network so it can model traffic the same way that the hosts see it.

The frag3 preprocessor is a target-based IP defragmentation module for Snort.

TCP REASSEMBLER

Once the IP layer is defragmented, Snort continues by reassembling TCP.

The TCP transport layer faces similar issues. TCP segments may arrive out of order or be duplicated, causing overlapping or retransmission of data that may be handled differently by different target operating systems. Incorrect TCP options can be handled differently by different TCP/IP implementations.

If a TCP segment is not processed or discarded at the destination host for any reason, but is processed by the IPS, the IPS could be vulnerable to an evasion attack.

The TCP stream reassembly module for Snort is called stream5. stream5 uses a “target-based” or “adaptive” approach to model TCP sessions in the same way that the destination operating system would process them.

The stream5 module also combines UDP packets together in a stream in the order that they are received so UDP traffic can be processed by Snort on a session basis.

PORTSCAN PROCESSOR

Now Snort searches for reconnaissance attacks. Portscan attackers scan using tools like nmap to determine what types of network protocols or services a target host supports. Most individual port queries sent by an attacker will be negative (meaning that the ports on the target machine are closed or unreachable.) The primary objective in detecting portscans is to detect and track negative responses.

The nmap tool and other scanners are very flexible in their portscan capabilities, and Snort needs the capability to watch for a variety of portscan attacks. Nmap can distribute attacks between different hosts, spoof the scanner’s source address, and run SYN scans that never complete their TCP connections. Snort’s portscan detector must also identify scan attempts that never return a response (perhaps because the response is being suppressed).

Snort detects portscan attempts through the use of the sfportscan preprocessor.

DETECTION ENGINE

PROTOCOL ANALYZER

The protocol analyzer normalizes traffic for inspection by Snort, ensuring that the correct areas are selected for inspection and the information is presented consistently.

Protocol Normalization

In the protocol normalization process, Snort preprocessors examine the traffic passed to them by the stream reassembler and identify the protocol’s components. They validate that the traffic conforms to a valid protocol specification.

Data Normalization

Data normalization is the process of putting the data used by a protocol into a single, consistent format. For instance, Unicode may be converted to ASCII, network encodings may be transformed to host-based encodings, or whitespace may be removed. This process is performed in the preprocessors.

Flow Analysis

Flow analysis classifies network application protocols into client and server data flows. 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.

RULESET SELECTOR

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 rulesets. This is done prior to any packet or stream processing. Once the Snort rules have been divided into rulesets, each incoming packet is matched to a corresponding ruleset based on the packet’s unique parameters.

MULTI-RULE SEARCH ENGINE

Snort then executes all the rules in the ruleset corresponding to the packet’s parameters. These rules may perform three different searches in the packet information:

- 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 INSPECTOR

If the multi-rule search engine finds matches, these matches are further inspected to fully validate that the corresponding rule indeed matches. 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

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 an assigned priority. This allows users to configure the amount of event information they feel is necessary for their particular networks. The resulting events are then sent to the Snort output system.

OUTPUT SYSTEM

The Snort output system receives events from the event selector and processes them. It checks whether they match suppression and thresholding rules, and withholds processing of these rules if they do. If not, the events are logged or passed to other systems for remediation and response purposes.

SUMMARY

By selecting relevant traffic, normalizing it, and matching it against a large ruleset, Snort's threat prevention components ensure that threats are caught and false positives and negatives are avoided, while maintaining performance. Sourcefire Intrusion Sensors combine Snort's threat prevention capabilities with enterprise-class administration, reporting, data management, and interoperability.

The Sourcefire 3D System, including Sourcefire Intrusion Sensors, RNA Sensors, and Defense Centers, unifies intrusion and vulnerability management technologies to provide customers with the most effective network security solution, against all threats, from all vectors, all the time.