jamk.fi

Threat detection of IPS in high load situation

Konsta Karttunen

Master's thesis
April 2018
School of Technology, Communications and Transport
Master's Degree Programme in Information technology
Cyber Security

jamk.fi

Description

Author(s)	Type of publication	Date
Karttunen Konsta	Master's thesis	4.2018
		Language of publication: English
	Number of pages 73	Permission for web publication: x

Title of publication

Threat detection of IPS in high load situations

Degree programme

Information Technology, Cyber Security

Supervisor(s)

Kotikoski Sampo

Assigned by

Konsta Karttunen

Abstract

Commercial network security solutions are typically measured by the throughput of the device in megabytes per second. Throughput is also quite often used to compare the different vendors and to size the security solution so that the solution is able to handle the network traffic passing through.

One method of avoiding a network security solution is to overwhelm the solution with excessive network traffic so that it either crashes or starts to bypass the traffic from the proper inspection. In a situation like this, more information about the performance of the solution would be extremely crucial.

The goal was to investigate the behavior of the security features, especially intrusion prevention systems, in high load situations. The target systems were loaded with high traffic volume using Ixia's BreakingPoint tool, and the Attack Pack feature of the tool was used to evaluate the security capabilities under the load. The difference between low and high load situations was also observed.

The results of the research revealed that different security solutions performed quite differently under heavy load. Additionally, it was noted that the measurement of the capabilities of a security product is not as straightforward as it might sound and comparing the results between different solutions poses a significant challenge.

Keywords/tags (subjects)

IPS, IDS, Ixia, Snort, Suricata

Miscellaneous

jamk.fi

Kuvailulehti

Tekijä(t)	Julkaisun laji	Päivämäärä
Karttunen Konsta	Opinnäytetyö, ylempi AMK	Huhtikuu 2018
	Sivumäärä 73	Julkaisun kieli Englanti
		Verkkojulkaisulupa myönnetty: x

Työn nimi
Threat detection of IPS in high load situations

Tutkinto-ohjelma
Information Technology, Cyber Security

Työn ohjaaja(t)
Sampo Kotikoski

Toimeksiantaja(t)
Konsta Karttunen

Tiivistelmä

Kaupallisia tietoverkkoihin suunniteltuja tietoturvakomponentteja vertaillaan tyypillisesti keskenään käyttäen laitteen tai tuotteen läpäisykykyä vertailtavana suureena. Läpäisykyky esitetään muodossa megatavuja sekunnissa ja tätä suuretta käytetään hyvin usein myös laitteen mitoituksessa niin, että mitoitettava laite kykenee suoriutumaan laitteen läpi kulkevasta tietoliikenteen määrästä.

Yksi tapa ohittaa tietoverkkoihin suunnatut tietoturvakomponentit on suunnata laitetta kohden tarpeeksi suuri määrä tietoliikennettä, jolloin laite voi esimerkiksi lakata toimimasta tai päästää liikennettä itsensä läpi ilman normaalissa kuormitustilanteessa tapahtuvaa liikenteen tarkempaa tarkastelua. Näissä tilanteissa tieto laitteen toiminnasta korkeassa kuormitustilanteessa on erittäin tärkeää.

Opinnäytetyön päämäärä oli tarkastella, kuinka IPS -laitteet käyttäytyvät äärimmäisen korkean kuormituksen tilanteissa. Tarkasteltaville laitteille aiheutettiin korkean kuormituksen tila käyttäen Ixian Breaking Point -työkalua. Kyseistä työkalua käytettiin myös haitallisen liikenteen lähettämiseen tarkasteltaville laitteille.

Tutkielman tulokset osoittivat eri tietoturvakomponenttien käyttäytyvän hyvin eri tavalla korkean kuormituksen tilanteessa. Samalla havaittiin, että tietoturvatuotteiden kyvykkyyksien tai suorituskyvyn mittaaminen on erittäin haastavaa ja yhteneväisten tulosten saaminen voi osoittautua erittäin hankalaksi.

Avainsanat (asiasanat)
IPS, IDS, Ixia, Snort, Suricata

Muut tiedot

Contents

Figures

Tables

1 Introduction

The purpose of the thesis is to investigate if the intrusion prevention solutions available on the market are able to detect and drop malicious traffic in high load scenarios. Ixia BreakingPoint test tool was used to test the commercial security solutions which house an intrusion prevention system.

1.1 Thesis background

Introduced over a decade ago, the first network intrusion prevention systems (IPS) were built on generic Intel servers with the purpose of blocking exploits that target vulnerable servers. Soon after, attacks against desktop clients emerged and the first generation of intrusion prevention struggled to maintain performance and security. This led to a new hardware-accelerated generation of IPS that could inspect much more traffic and at higher speeds than software-only solutions. (NSS Labs 2017, 4)

Firewall technology is one of the largest and most mature security markets. Firewalls have undergone several stages of development, from early packet filtering and circuit relay firewalls to application layer (proxy-based) and dynamic packet filtering firewalls. Throughout their history, however, the goal has been to enforce an access control policy between two networks, and they should therefore be viewed as an implementation of policy. (NSS Labs 2016, 5)

A firewall is a mechanism used to protect a trusted network from an untrusted network, while allowing authorized communications to pass from one side to the other, thus facilitating secure business use of the Internet. With the emergence of new web applications and security threats, however, firewalls are evolving further. Next generation firewalls (NGFWs) traditionally have been deployed to defend the network on the edge, enterprises have expanded deployment options to include internal segmentation. (NSS Labs 2016, 5)

These security solutions are the backbone of the network security in most of the networks, which makes them an extremely critical point in the network and sets quite high standards when it comes to performance and reliability.

1.2 Research problem

Commercial network security solutions are typically measured by the throughput of the device in megabytes per second. Additionally, throughput is quite often used to compare the different vendors and to size the security solution so that the solution is able to handle the network traffic passing through.

One method of avoiding a network security solution is to overwhelm the solution with excessive network traffic so that it either crashes or starts to bypass the traffic from the proper inspection. In a situation like this, more information about the performance of the solution would be extremely crucial.

The objective of this research is to study how the handling of the malicious traffic is performed when the security solution is experiencing a very high load. Currently, benchmarking providers such as NSS Labs perform under-load tests so that the overall system load of the solution under test is about 80% of the maximum.

1.3 Research methods and challenges

The primary objective in the research was to study how the handling of malicious traffic in the different intrusion prevention systems, either embedded into a security solution such as a next generation firewall or as a standalone system such as Snort, is performed when the system load is over 90%.

The research aims to compare the amount of malicious traffic propagating through the tested system in normal and high load situations.

Before the implementation phase it was unknown if the target devices can be taxed up to 90% of the overall system capacity. Additionally, the full functionality of the tool used for testing could not be confirmed before the actual implementation phase. To avoid problems with the testing, following decisions were made:

- Hardware devices under the test were low end devices with minimal intrusion prevention performance.
- Virtual devices were only given minimum amount of CPU power.
- The test environment should be able to utilize 1 Gigabit traffic volume.

A case study type of approach was selected as the research method as it is a method used to narrow down a very broad field of research into one easily researchable topic. (Shuttleworth, M. 2008)

A mix of qualitative and quantitative research is also included as the results are to consist of measurable data such as the performance numbers of IPS systems and non-measurable information such as the behavior of the devices under high load.

2 IDS and IPS

Intrusion detection and prevention systems are security controls designed to monitor network traffic and to filter the malicious traffic off from the network. These systems can be used as a separate device in the network or as an integrated part of a security platform like a firewall.

2.1 History and development of IPS

Both of these systems have their actual roots in auditing as in 1980, James Anderson wrote a technical report called Computer Security Threat Monitoring and Surveillance for the U.S. Air Force. The paper showed that audit records could be used to help identify computer misuse and identify threat classifications, and it offered suggestions to improve auditing of systems to identify misuse. (Trost 2009)

The older of these two systems, IDS was developed heavily during 1980s and 1990s. The first IPS systems started to emerge in the late 1990s, most notably SNORT, an open source libpcap-based packet sniffer and logger, developed by Marty Roesch, which is today the de-facto standard in intrusion prevention.

The main difference between an intrusion detection system and intrusion prevention system is that the intrusion prevention system is able to drop or modify the traffic passing through.

2.2 IDS

An intrusion detection system (IDS) is a security control or countermeasure that has the capability to detect misuse and abuse of, and unauthorized access to, network

resources (Adesina, Barker & Burns 2012). Intrusion detection systems do not take any action when malicious traffic is seen in the network. Figure 1 presents a typical IDS deployment scenario.

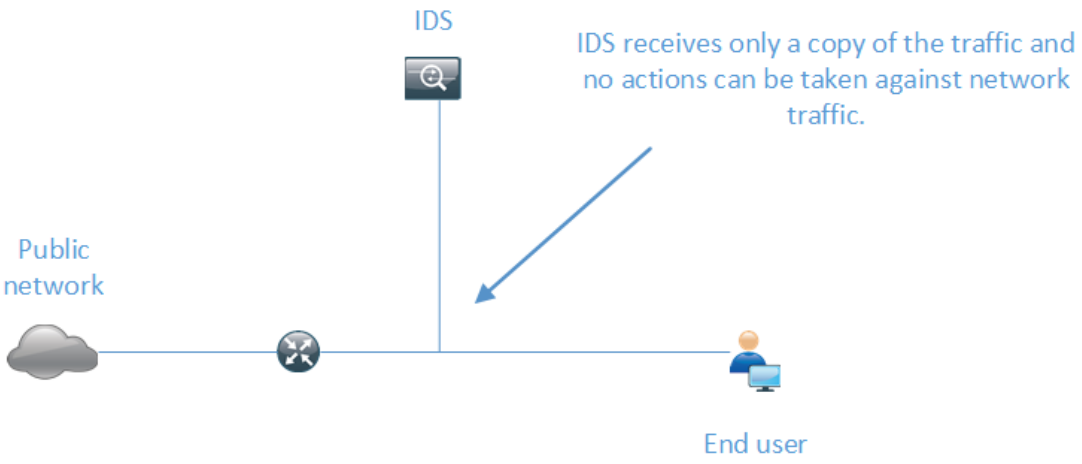


Figure 1 Example IDS deployment

An IDS needs only to detect threats and as such is placed out-of-band on the network infrastructure, meaning that it is not in the true real-time communication path between the sender and receiver of information. Rather, IDS solutions will often take advantage of a TAP or SPAN port to analyze a copy of the inline traffic stream (and thus ensuring that IDS does not impact inline network performance). (Palo Alto Networks 2017)

IDS was originally developed this way because at the time, the depth of analysis required for intrusion detection could not be performed at a speed that could keep pace with components on the direct communications path of the network infrastructure. (Palo Alto Networks 2017)

2.3 IPS

A security control or countermeasure that has the capability to detect and prevent misuse and abuse of, and unauthorized access to, networked resources is an intrusion prevention system. (Adesina et al. 2012)

The main capability provided by Network Intrusion Prevention is the ability to prevent malicious traffic from reaching the target system (Hogue & Carter 2006). Figure 2 presents an IPS deployment scenario.

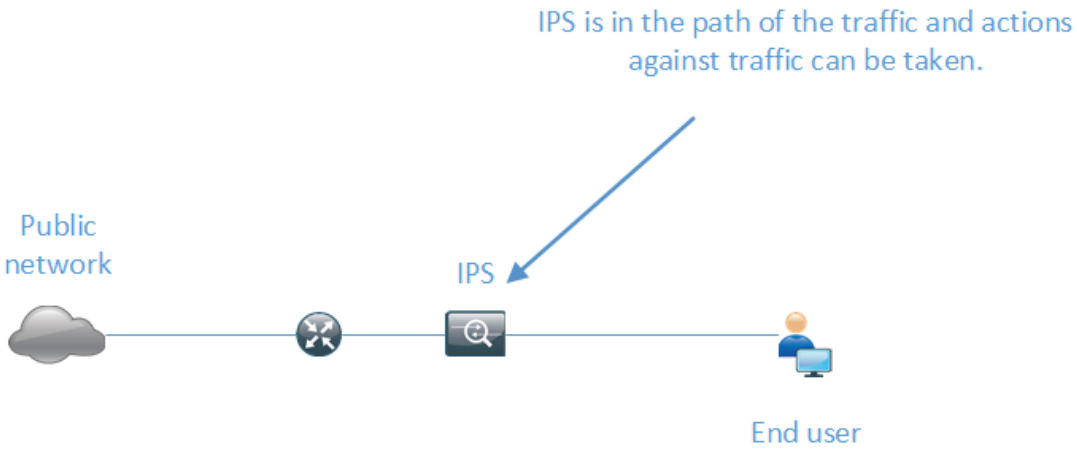


Figure 2 Example IPS deployment

A wide number of platforms can host an IPS. IPS can be deployed as a dedicated appliance or in a combination with other security controls such as stateful packet filtering as deployed in a firewall. IPSs can also be found in host computers usually as a part of the endpoint protection suite products.

IPS was originally built and released as a standalone device in the mid-2000s. This, however, was in the advent of today's implementations, which are now commonly integrated into Unified Threat Management (UTM) solutions (for small and medium size companies) and next-generation firewalls (at the enterprise level). (Palo Alto Networks 2017)

Modern wireless networks can also house an IPS system capable of defeating attacks targeted against wireless users.

Network Intrusion Prevention provides a proactive component that effectively integrates into overall network security framework. Combining Network Intrusion Prevention with other security components, such as a Host Intrusion Prevention System (HIPS), an Intrusion Detection System (IDS), and perimeter firewalls, provides a robust defense-in-depth network security solution. (Hogue & Carter 2006)

Network IPS operates at layer three and four of the Open systems interconnect model. A network IPS has four main features: (Adesina et al. 2012)

A network IPS can detect attacks on several different types of operating systems and applications, depending on the extent of its database.

1. A single device can analyze traffic for a large scale of hosts on the network, which makes network IPSs a cost-effective solution that decreases the cost of maintenance and deployment.
2. As sensors observe events from and to various hosts and different parts of the network, they can correlate the events, hosts, and networks to higher-level information. In conjunction with the correlation, they can obtain deeper knowledge of malicious activity and act accordingly.
3. A network IPS can remain invisible to the attacker through a dedicated interface that monitors only network traffic and is unresponsive to various triggers or stimuli.

There are three types of approaches how the network IPSs are investigating the traffic passing through: signature based, anomaly based and policy based approach.

A network IPS that analyzes network traffic and compares the data in the flow against a database of known attack signatures is called signature-based IPS. A signature-based IPS looks at the packet headers and/or data payloads when analyzing network traffic. All signature-based IPSs require regular updates for their signature databases. (Adesina, et al. 2012)

A network IPS that analyzes or observes network traffic and acts if a network event outside normal network behavior is detected is called anomaly-based IPS. The two types of anomaly-based network IPSs are statistical anomaly detection and protocol verification. (ibid)

A network IPS that analyzes traffic and acts if it detects a network event outside a traffic policy is called policy based IPS. A traffic policy usually involves permitted or denied communications over a network segment similar to an enterprise-class firewall (ibid). Figure 3 shows a Cisco network IPS management console.

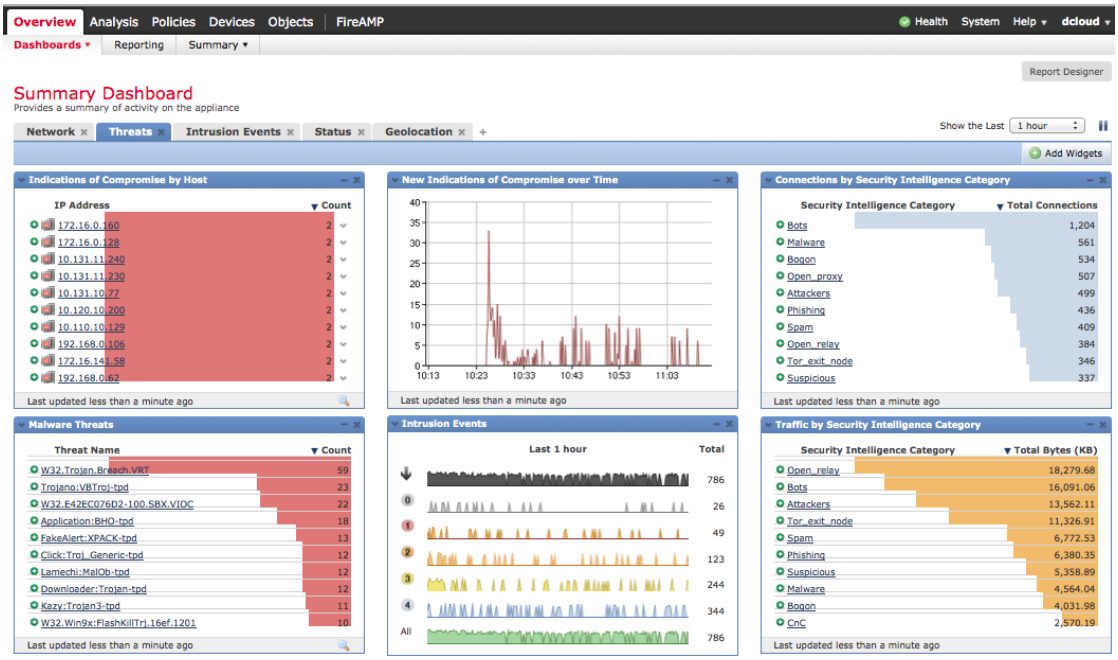


Figure 3 Cisco Firepower Management Center

Another form of intrusion prevention is the host IPS (HIPS). Often referred to as endpoint security controls, a HIPS consists of operating system security controls or security agent software installed on hosts that can include desktops PCs, laptops, or servers. Host IPSs in most cases extend the native security controls protecting an operating system or its applications (Adesina, et al. 2012).

F-secure SAFE or Sophos endpoint security are good examples of an endpoint protection suite which also incorporates host IPS. Figure 4 shows an advanced HIPS solution. Figure 4 shows an example of a host based IPS solution.

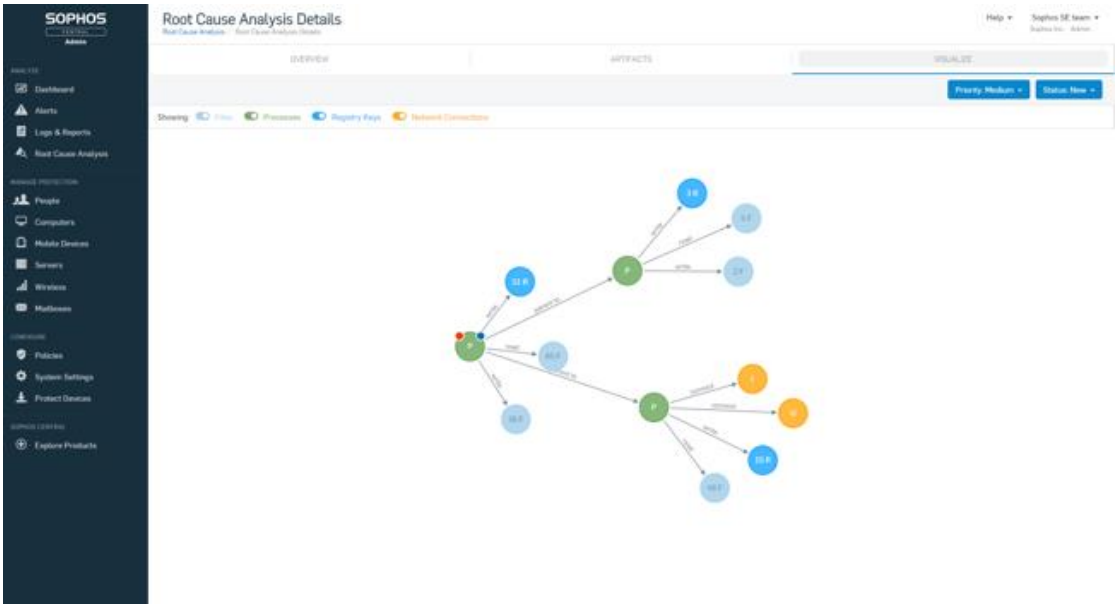


Figure 4 Sophos HIDS root cause analysis (www.sophos.com)

A Wireless IPS (WIPS) solution can be found on almost every commercial wireless solution on the market. However, the capabilities of the IPSs vary a great deal from simple rogue access point detection to active jamming of the rogue access points. The main capabilities of a wireless IPS are rogue detection, wired containment and wireless containment.

In rogue detection, the is few different methods for determining that an access point is connected to the wire. The most basic is a +/- 1MAC address check of traffic that has been on the wire and seen wirelessly. If wired traffic is observed with a MAC address that is within 1 of wireless traffic, that device will be tagged as a wired connected rogue. (Aruba Networks 2014)

Wired containment is performed by ARP poisoning the default gateway of a rogue device connected to the wire. The detecting AP or AM will perform the containment. The wirelessly detecting device needs to be on the same VLAN as the rogue for the wired containment to be successful. (Aruba Networks 2014)

There are two types of wireless containment, deauth and tarpitting. Both start out the same way. The access point will send de-authentication packets to the access point and the client device. Most client devices will automatically try to reconnect to the network. When deauth is selected, the access point will send another deauth packet once the client is connected to the network. With modern clients that can

happen as quickly as every 15 milliseconds. Tar-pitting will behave a little differently. When the client device attempts to reconnect to the network, the access point will respond with a probe response that has some fake data in it to induce the client device to connect to the access point rather than the rogue device. The client device then takes some time to realize the connection isn't going anywhere. At that point, it disconnects and starts over. (Aruba Networks 2014)

3 IPS terminology

Security controls in IPS systems detect and produce alerts from a number of factors. Security controls are divided into a four-different situation category by the type of the detection. A detection can be, for example, from a legitimate malicious traffic or from a misconfiguration of an IPS system.

3.1 Security controls

IPS security controls are situations which describes how alarm or response against detection is classified. Security controls are classified in one of the following terms: (Adesina, et al. 2012)

True positive

A situation in which a signature fires correctly when intrusive traffic for that signature is detected on the network. The signature correctly identifies an attack against the network. This represents normal and optimal operation. (Adesina, et al. 2012)

False positive

A situation in which normal user activity triggers an alarm or response. This is a consequence of non-malicious activity. This represents an error and generally is caused by excessively tight proactive controls or excessively relaxed reactive controls. (Adesina, et al. 2012)

True negative

A situation in which a signature does not fire during normal user traffic on the network. The security control has not acted and there was no malicious activity. This represents normal and optimal operation. (Adesina, et al. 2012)

False negative

A situation in which a detection system fails to detect intrusive traffic although there is a signature designed to catch the activity. In this situation, there was malicious activity, but the security control did not act. This represents an error and generally is caused by excessively relaxed proactive controls or excessively tight reactive controls. (Adesina, et al. 2012) Table 1 summarises IPS action situations.

Table 1 IPS Signature summary

	POSITIVE	NEGATIVE
TRUE	True Positive: Alerted on intrusion attemp	True Negative: Not alerted on benign activity
FALSE	False Positive: Alerted on benign activity	False Negative: Not alerted on intrusion attemp

3.2 Signatures

All IPS products use signatures. Signatures are the means of an IPS to prevent the malicious activities, where it is happening on a host or in a network. Simply put IPS signature is any distinctive characteristic that identifies something. (Hogue & Carter 2006)

IPS Signatures can be distinguished by signature type, action or trigger. Signatures fall into one of the following two basic categories depending on their functionality.(ibid.)

3.2.1 Atomic Signatures

Atomic signatures represent the simplest signature type. For an atomic signature, a single packet, activity, or event is examined to determine if the signature should trigger a signature action. Because these signatures trigger on a single event, they do

not require intrusion system to maintain state. The entire inspection can be accomplished in an atomic operation that does not require any knowledge of past or future activities. (Hogue & Carter 2006)

State refers to situations in which you need to analyze multiple pieces of information that are not available at the same time. It also refers to tracking established TCP connections (connections that have gone through the initial three-way handshake). Valid TCP traffic also refers to traffic that has the correct sequence numbers for an established connection. For Network IPSs, state signatures usually refer to signatures that require analyzing traffic from multiple packets. (ibid.)

A good example of a network based atomic signature is ARP (address resolution protocol) spoofing attack. This attack can be detected by inspecting a single packet and as everything is contained in a single packet, no state information is needed to identify the attack.

An ARP spoofing attack is an attack where an attacker floods a network with spoofed ARP information in order to divert the traffic destined to the network's default gateway to attacker.

3.2.2 Statefull signatures

Unlike atomic signatures, stateful signatures trigger on a sequence of specific events that requires the IPS device to maintain state. The length of time that the signatures must maintain state is known as the event horizon. Configuring the length of the event horizon is a tradeoff between consuming system resources and being able to detect an attack that occurs over a long period of time. (Hogue & Carter 2006)

Stateful signatures usually require several pieces of data to match an attack signature. The maximum amount of time over which an attack signature can successfully be detected (from the initial data piece to the final data piece needed to complete the attack signature) is known as the event horizon. The intrusion system must maintain state information for the duration of the event horizon. The length of event horizon varies from one signature to another. (ibid.)

Often, Network-based IPS signatures are stateful signatures because the information needed can usually be distributed across multiple packets. Even a simple string

match signature is usually stateful because the string can occur across multiple packets (because the IPS must examine the data from all the packets until the successful match is made). For example, if you want to search for the string /etc/password in an HTTP URL, you might have to check multiple packets because the string can be distributed across more than one packet. (Hogue & Carter 2006)

3.3 Signature triggers

The heart of any IPS signature is the mechanism that causes it to trigger. These triggering mechanisms can be simple or complex, and every IPS incorporates signatures that use one or more of these basic triggering mechanisms to trigger signature actions. These triggering mechanisms can be applied to both atomic and stateful signatures. (ibid.)

The most commonly used triggering mechanisms today are pattern detection, anomaly-based detection and behavior based detection.

The simplest triggering mechanism is identifying a specific pattern. This pattern can represent a textual or binary string or it can be other pattern, such as a sequence of function calls. (ibid.)

Regular expression (REGEX) patterns are the most commonly used type of a pattern detection. Specifying string patterns using regex provides the ability to efficiently search for textual patterns (using a single regular expression) while making it harder to bypass the pattern without detection. (ibid.)

For example, the following regex string searches for an attempt to change the working directory to the root directory during an FTP session (Hogue & Carter 2006):

[\t]*[Cc][Ww][Dd][\t]+[~]root

Anomaly-based (also known as profile-based detection) signatures are not based on a specific event. Instead. these signatures trigger when a certain activity deviates from what is considered normal.

In order to utilize anomaly based detection, a certain baseline must be established. In other words, the baseline refers to what is normal and everything outside that scope can be flagged as anomaly.

Behavior-based detection is similar to pattern detection, but instead of trying to define specific patterns, you are defining behaviors that are suspicious based on historical analysis. The behaviors define classes of activity that are known to be suspicious. (Hogue & Carter 2006)

3.3.1 Considerations

Hogue & Carter (2006) state that "*one drawback with atomic signatures is that you have to know all the atomic events that you want to look for. For each of these events, you then have to create the appropriate signature. As the number of atomic signatures increases, just managing the different signatures can become overwhelming*." This could easily lead to a very large signature tables which can lead to a high latency when searching for the correct event.

Hogue & Carter (2006) also state that "*the main limitation to stateful signatures is that maintaining state consumes memory resources on your IPS/IDS device. Usually, however, this is not a significant problem if the IPS product is designed to efficiently use its resources. If your IPS does not efficiently manage resources when maintaining state, then the large consumption of resources (such as memory and CPU) can lead to a slow response time, dropped packets, missed signatures, and so on, which adversely impacts the effectiveness of your IPS*."

3.4 Snort and Suricata

Snort and Suricata are both IPS software which can be found in commercial security solutions. Both can be deployed as IDS or IPS sensor in the network.

Snort is an open source network intrusion prevention system (IPS) by Cisco. It is capable of performing real-time traffic analysis and packet logging on IP networks. It can perform protocol analysis, content searching and matching, and detect a variety of attacks and probes. Snort can be used as a straight packet sniffer like tcpdump, a packet logger (useful for network traffic debugging), or as a full-blown network intrusion prevention system. (Snort 2016)

Three possible configuration modes for Snort are:

- Sniffer mode, which reads the packets off of the network and displays them in a continuous stream on the console (Snort 2016)
- Packet Logger mode, which logs the packets to disk. (Snort 2016)
- Network Intrusion Detection System (NIDS) mode, which performs detection and analysis on network traffic (Snort 2016)

Snort also uses a concept of preprocessors, which allow the functionality of Snort to be extended by allowing users and programmers to drop modular plugins into Snort fairly easily. Preprocessor code is run before the detection engine is called, but after the packet has been decoded. The packet can be modified or analyzed in an out-of-band manner using this mechanism. (Snort 2016).

A good excample for a Snort preprocesssor would be the sfPortscan preprocessor. The sfPortscan module, developed by Sourcefire, is designed to detect the first phase in a network attack: Reconnaissance. (ibid.) In the Reconnaissance phase, an attacker determines what types of network protocols or services a host supports.

Snort uses a simple description language. 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. 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. (ibid.) Figure 5 presents a sample snort rule.

```
alert tcp any any -> 192.168.1.0/24 111 \
    (content:"|00 01 86 a5|"; msg:"mountd access";)
```

Figure 5 A sample snort rule

Snort can be considered as de facto standard in intrusion prevention and it is used by most of the security vendors to provide the intrusion prevention capabilities in their solutions.

Suricata is a high-performance Network IDS, IPS and Network Security Monitoring engine. It is open source and owned by a community-run non-profit foundation, the Open Information Security Foundation (OISF). Suricata is developed by the OISF. (Suricata 2016)

Suricata implements a complete signature language to match on known threats, policy violations and malicious behavior. Suricata will also detect many anomalies in the traffic it inspects. Suricata is capable of using the specialized Emerging Threats Suricata ruleset and the VRT ruleset. (Suricata 2016)

Suricata is natively multithreaded and therefore, a single Suricata instance is capable of inspecting multi-gigabit traffic.

3.5 High performance intrusion prevention

As the scale of the traffic flowing through IPS systems has increased rapidly over the years and the amount of different attacks is increasing, the IPS systems face a serious challenge in terms of detection rate and throughput. For example, a signature-based IPS, such as Snort, employs thousands of rules that contain intrusion patterns.

Each Snort rule is divided into two logical sections: the rule header and the rule options. The rule header contains the rule's action and a classification filter that consists of five fixed fields: protocol, source IP address, source port, destination IP address, and destination port. The rule option contains alert messages and pattern information on how a packet payload should be inspected. (Panthan 2014)

The results of the header classification identify the related rule options that will be checked in the follow-up deep packet inspection (DPI). Deep packet inspection is based on pattern matching, in which Snort employs two types of patterns: strings and regular expressions. Both header classification and deep packet inspection as the core functions of NIDS are computation-intensive, which has challenged the conventional computing architectures with demanding CPU, memory, and I/O requirements. (ibid.)

When examining today's hardware techniques, we have general-purpose processors, such as CPUs (central processing units) and GPUs (graphics processing units), on one end of the spectrum and application-specific integrated circuits (ASICs) on the other. (Panthan 2014)

ASICs provide the best performance but require a complete and extremely expensive re-fabrication of the circuits. Fortunately, there exists some architecture between

these two extremes. Reconfigurable hardware, such as a field-programmable gate array (FPGA), offers the best of both worlds. Modern FPGAs provide superior performance, and they can be reprogrammed on the fly. (ibid)

Hardware vise packet header classification can be approached with two different methods. Ternary content addressable memory (TCAM) is a specialized ASIC widely used in network search engines. Most of the existing multi-match packet classification engines are based on TCAMs in which each input performs a parallel search over all entries in one clock cycle, and only the first matching index is output. (ibid.)

TCAMs are expensive and not scalable with respect to clock rate, power consumption, or circuit area, compared to static random access memories (SRAMs). As the rule set size increases rapidly, alternate hardware platforms are needed for multi-match packet classification engines. State-of-the-art SRAM-based field programmable gate array (FPGA) devices provide a high clock rate and a large amount of on-chip dual-port memory with configurable word width. It takes a few milliseconds to reconfigure an entire FPGA, and the update frequency of NIDS rules is on the order of days. Thus, FPGA has become an attractive platform for realizing real-time network processing engines. (Panthan 2014)

The functions of NIDS rely on multi-pattern string matching, which scans the input stream to find all occurrences of a predefined set of string-based patterns rather than a single pattern. Due to the explosive growth of network traffic, multi-pattern string matching has been a major performance bottleneck in NIDS, which has to scan the incoming traffic in real time on fast links (e.g., 100 Gbps Ethernet and beyond). For example, it has been reported that the string matching time accounts for 40% to 70% of the Snort running time. Simple and efficient hardware-based multi-pattern string matching engines have become a necessity for high-speed NIDS. (ibid.)

String matching has been a classic problem for decades. According to the implementation platform, the state-of-the-art solutions can be generally divided into three categories: multi-core processor–based, application-specific integrated circuit (ASIC)–based, and field programmable gate array (FPGA) based solutions. Each of the three hardware solutions has its own pros and cons. (Panthan 2014)

Advanced multi-core processor based solutions can improve the aggregate throughput dramatically by using a large number of threads to process multiple input streams in parallel. On the other hand, it has been observed that the memory access pattern in string matching is irregular. This results in relatively low per-stream throughput, which is critical for real-time network traffic processing. Although it is possible to split an input stream into several sub-streams with partial overlap among the sub-streams, additional complexity is introduced in scheduling, buffering, and ordering. (ibid.)

ASIC–based solutions provide impressively high per-stream throughput while their applicability is limited by the high implementation cost and low reprogrammability. Combining the flexibility of software and the near-ASIC performance, FPGA technology has become an attractive option for implementing high-performance string matching engines. (ibid.)

Regular expression (regex) matching is an important mechanism used by modern IPS, such as Snort to perform deep packet inspection against potential threats. Due to the large number of patterns to scan for and the increasing bandwidth of network traffic, regular expression matching is becoming not just a bottleneck but itself a vulnerability of the NIDS. (Panthan 2014)

Network intrusion detection systems (NIDS) have been widely used to secure the networks. However, the performance of NIDS must be capable of catching up with the explosive growth of network traffic to prevent the NIDS itself becoming the target of attacks. The core functions of modern NIDS include multi-match packet classification and deep packet inspection, which is based on multi-pattern string matching and regular expression matching. These functions are computation-intensive, especially when the size of the NIDS rule set is large. This has challenged conventional computing architectures with demanding CPU, memory, and I/O requirements. Dedicated hardware accelerators become a necessity to address these challenges. (Panthan 2014)

4 Test scenario

In the test phase, the solutions under test are flooded with network traffic. This traffic simulates the normal network traffic flowing through the devices every day in live networks. When the overall system load level is near maximum for each solution, malicious traffic is added into the traffic pattern. This simulates the situation where an attacker may try to overwhelm the security device to bypass the system exploiting the congestion algorithms applied in high load situations.

The BreakingPoint test environment consists of Ixia virtual controller, virtual blade and a virtual test switch. Virtual controller performs the management functions for the test environment. Virtual blade is attached to the controller and provides the actual functionalities in the test environment such as traffic generation. Virtual switch is used as a control channel between the virtual controller and the virtual blade and it also functions as a connection point for physical devices under the test. Figure 6 presents an example deployment scenario of the BreakingPoint test tool.

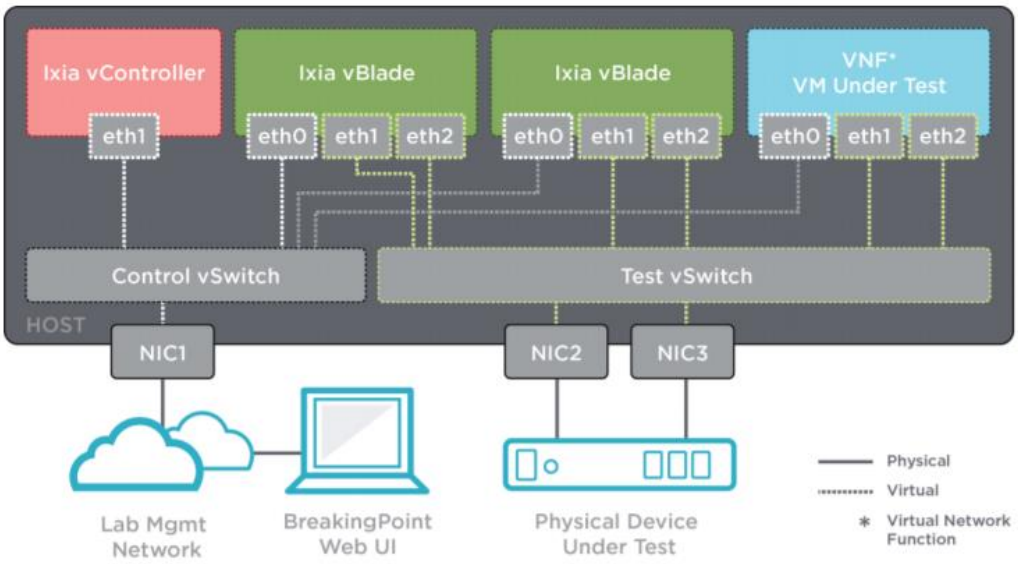


Figure 6 BreakingPoint VE deployment for both virtual and physical device tests (Ixia)

The test environment consists of Ixia BreakingPoint solution virtualized in VMware environment. The tested systems are deployed between the Breaking point using two network interfaces which are used as source and destination for the simulated traffic. Figure 7 presents VMware environment's virtual networking for BreakingPoint.

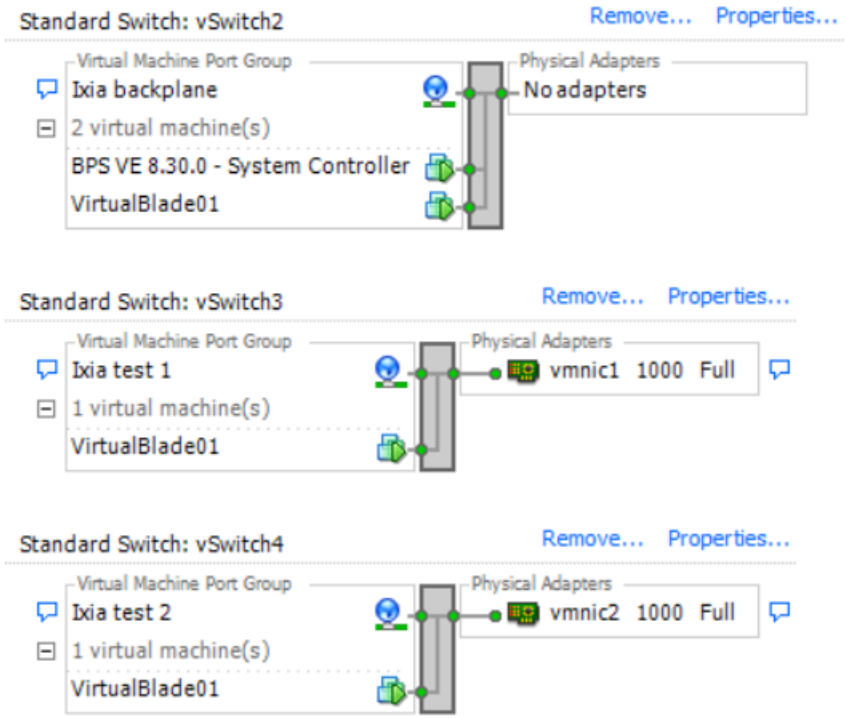


Figure 7 VMware virtual networking in test environment

The virtual controller software is deployed as open virtualization format template (OVF) on a VMware host. After the deployment, the virtual blade is deployed using the controller software. Figure 8 presents the virtual blade deployment process.

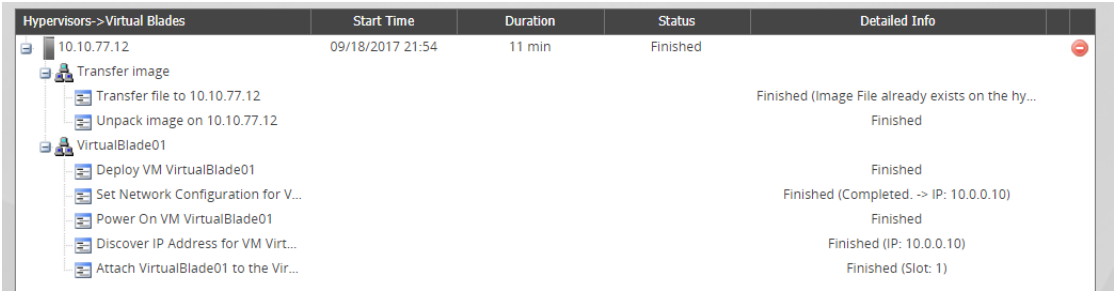


Figure 8 Virtual blade deployment example

After the deployment of the virtual blade, the BreakingPoint software can be launched from the virtual controller. Figure 9 shows the main page of the BreakingPoint software.

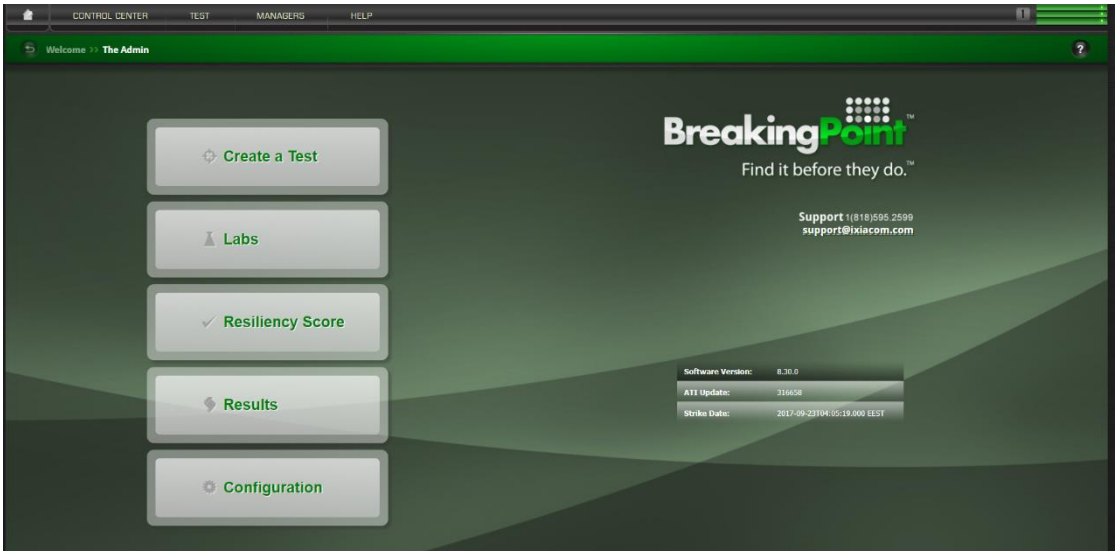


Figure 9 BreakingPoint main menu

BreakingPoint test consists of two main components. These components are network neighborhood and test components. Network neighborhood includes the addressing rules available for each test interface. Network neighborhood has different components which break the neighborhood into a smaller subset which is combined with tags. As seen in Figure 10, the neighborhood used in the test has interface, router and host objects.

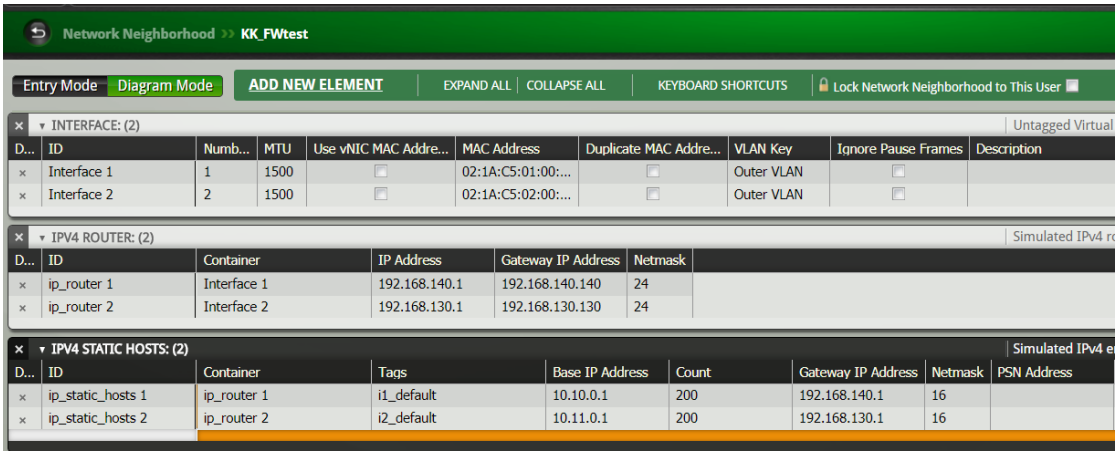


Figure 10 Test environments virtual network neighborhood

Interfaces are the interfaces of the virtual blade attached into the VMware virtual switch seen in Figure 8. Ipv4 router and ipv4 static hosts. The objects present the networks inside the virtual blade which are used for traffic generation. In the neighborhood shown in Figure 11, the generated traffic for tests is flowing from host network 10.10.0.0 /16 towards 10.11.0.0 /16 network where the simulated services reside.

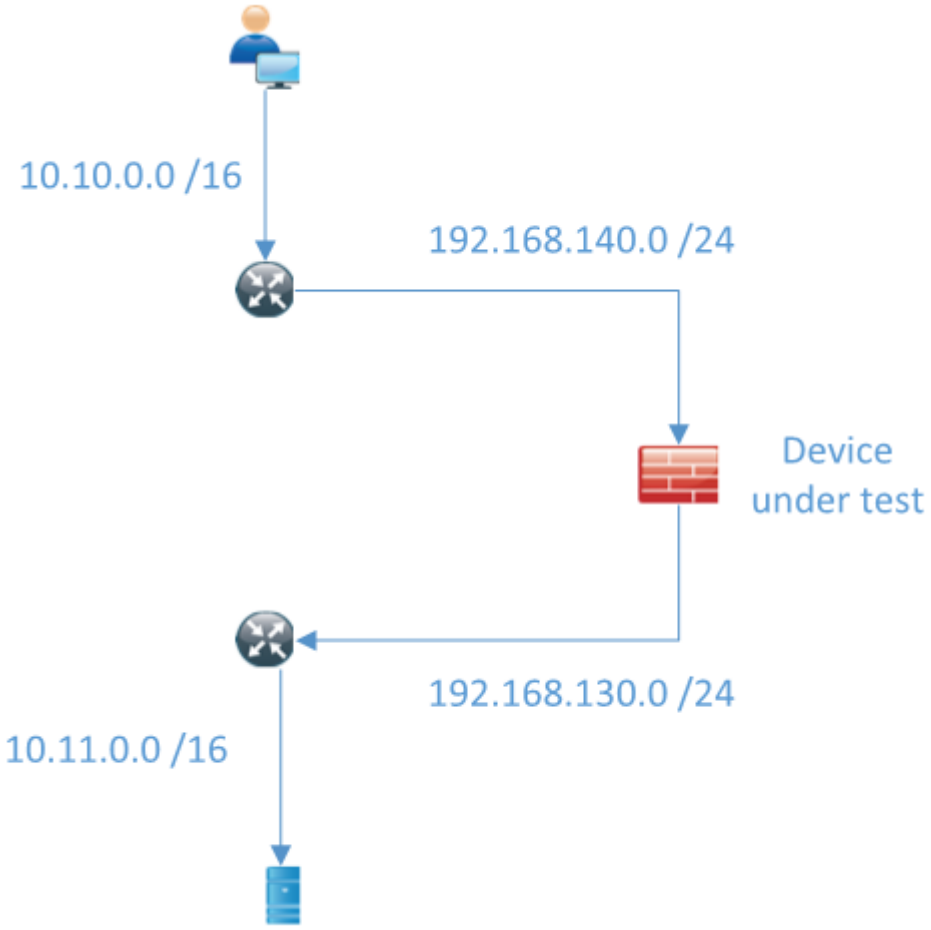


Figure 11 Network neighborhood for tests

From the test components section two options are used for the tests. These options are application simulator and security. The application simulator defines the traffic pattern used to load down the devices under test. Security enables to include harmful traffic into the test pattern.

The main parameters in the application simulator for the tests are application profile determining the traffic pattern and load profile determining the amount of traffic sent in different phases of the test.

Two different BreakingPoint default application profiles were chosen for the tests, NGFW Enterprise Perimeter Traffic Mix 2016 seen in Figure 12 which emulates the enterprise perimeter protocol mix described in a 2016 Next Generation Firewall Test

Methodology report (BreakingPoint help).

Name	Weight	Seed	Sessions	% Bandwidth	% Flows	# Bytes
Youtube October 2011 (Deprecated)	1,159	Generated	6	11.01	0.77	2,296,980
Bandwidth HTTP	1,869	Generated	1	17.76	4.90	585,745
Bandwidth BitTorrent File Download	1,082	Generated	2	10.28	2.83	586,792
Facebook	580	Generated	1	5.51	1.47	604,165
HTTPS Simulated 512k Data	966	Generated	2	9.18	2.58	573,502
Raw 2.5m Data	80	Generated	1	0.76	0.02	5,120,504
FTP Enterprise	500	Generated	3	4.75	2.73	281,441
GTalk Voicecall	966	Generated	6	9.18	24.19	61,270
AOL Instant Messenger	116	Generated	1	1.10	4.04	44,014
Yahoo Mail	966	Generated	5	9.18	14.03	105,664
Twitter View Favorites	309	Generated	3	2.94	26.32	18,017
Amazon S3 Retrieve Objects	773	Generated	2	7.34	0.93	1,270,312
SMTP Email	193	Generated	2	1.83	14.63	20,239
Gmailclassic_130508	966	Generated	23	9.18	0.55	2,694,342

Figure 12 NGFW Enterprise Perimeter Traffic Mix 2016 pattern

IPS Core Traffic profile can be seen in Figure 13 where the traffic comprises protocols SSH, RSTP and SMTP (BreakingPoint help). According to Ixia, the IPS Core Traffic profile is ideally designed to test IPS systems.

Name	Weight	Seed	Sessions	% Bandwidth	% Flows	# Bytes
SMB Client File Download	140	3311365488	2	14.58	15.22	23,732
PostgreSQL	100	Generated	2	10.41	10.87	3,590
DCE RPC	40	Generated	1	8.33	4.35	636
FTP	30	Generated	5	1.25	3.26	12,254
SMB Null Session	30	Generated	2	3.12	3.26	2,481
AOL Instant Messenger	20	Generated	1	4.17	2.17	13,589
SIP/RTP Direct Voice Call (TCP Transport)	10	Generated	4	0.52	1.09	227,058
NFSv3	10	Generated	3	0.69	1.09	2,614
SSH	10	Generated	1	2.08	1.09	6,921
RTSP	10	Generated	3	0.69	1.09	8,639
SMTP Email	120	Generated	2	12.50	13.04	1,512
IPS - Images	20	Generated	2	2.08	2.17	123,999
IPS - Small Images	100	Generated	2	10.41	10.87	65,036
IPS - HTTP Text	240	Generated	2	24.99	26.09	1,954,845
IPS - HTTP Video	20	Generated	2	2.08	2.17	1,433,254
IPS - HTTP Audio	20	Generated	2	2.08	2.17	9,357,767

Figure 13 IPS Core Traffic pattern

The load profile in the application simulator defines the behavior of traffic during the different phases of the test. In BreakingPoint, a load profile consists of three phases: ramp up, steady state and ramp down phase.

In the ramp up phase the BreakingPoint system opens as many TCP connections as possible between the source and destination networks seen in Figure 11. The steady state is used to open and close the TCP sessions opened in the ramp up phase. Opened connections are used for the traffic generator to produce traffic for the test defined in the application profile. In the ramp down phase, TCP connections are closed and no new sessions are opened. Table 2 shows the selected values for the tests. Appendix 1,2 and 3 describe variables available for the load profile phases.

Table 2 BreakingPoint load profile settings.

Parameter	Description	Valid values	Description
Ramp Up Behavior	Sets how the component will handle sessions during different test phases	Full Open	The full TCP handshake performed when sessions are opened.
Steady-State Behavior		Open and Close Sessions	Sessions are closed as they finish sending data, and new sessions are opened.
Ramp Down Behavior		Full Close	The full TCP session close is performed.

The security section defines the malicious traffic used in tests. A premade strike list, which houses the number of strikes is used. Additionally, any advanced IPS evasion is not used in order to simplify the tests. A strike list named Strike Level 1 – 2017 includes 21 strikes from year 2017 with CVSS score 10.0. This strike list is used against the devices under test.

A single strike which the tested IPS system can block is also used. This strike is chosen per device basis. The strike lists have a delayed start of 10 seconds, which gives the load profile enough time to ramp up the simulated traffic.

Many more advanced parameters are available in the BreakingPoint test system. These options are outside the scope of this research because the research focuses mainly on the IPS performance. For every tested device, a certain point of where the device cannot handle the traffic must be determined before the test with the malicious traffic can be performed. Otherwise, the high load scenario cannot be reached.

The problem with reaching the high load scenario was solved by high over subscripting the amount of traffic in the test network. The number of virtual hosts and services was set way too high for the system under the test to handle, which caused the system load of the tested system to rise high enough so that the tests could be executed.

It was also considered how to perform the over subscripting. If the focus was only on raising the amount of TCP connections through the tested system, the traffic drops would became totally random as the device runs out of memory. Instead of a high

amount of connections, the volume of traffic in megabytes was set as the major component.

5 System tests

The total of three different systems were tested during the test period. The systems including snort intrusion prevention system were Sophos UTM firewall and Cisco ASA Firepower 5506 firewall. Suricata intrusion prevention system was included in OPNsense system which is a fork from PFSense open source firewall.

5.1 Test scenario

The first test run included only attacks from Strike Level 1 – 2017 attack pack without any background traffic. The purpose of the test was to set a base line in how many attacks the target device was able to prevent. Some of the attacks had many iterations so the total number of attacks in the tests was 210

The second test run combined the Strike Level 1 – 2017 attack pack with NGFW Enterprise Perimeter Traffic Mix 2016 traffic pattern and the third test used IPS Core Traffic pattern with Strike Level 1 – 2017 attack pack. The fourth test included only one attack which the device under test was able to detect combined with IPS Core Traffic pattern.

IPS process CPU utilization was monitored with “top” command. In Sophos UTM and OPNsense it was possible to monitor the IPS process CPU usage continuously; however, in Cisco Asa Firepower the top command had no options for continuous monitoring and therefore, the exact utilization of the IPS process could not be obtained. Figure 14 shows an example of a top output from Cisco Asa Firepower.

```
top - 19:22:13 up 31 days,  5:58,  1 user,  load average: 13.96, 13.80, 13.72
Tasks: 159 total,   1 running, 158 sleeping,   0 stopped,   0 zombie
Cpu(s): 30.8%us,  6.8%sy,  0.0%ni, 62.3%id,  0.1%wa,  0.0%hi,  0.0%si,  0.0%st
Mem:   3921756k total,  2961616k used,   227268k free,    65128k buffers
Swap:  3996668k total,   265544k used,  3731124k free,   667744k cached

  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
 5207 root       0 -20 1140m 513m  89m S   99 13.4  44976:43 lina
16836 root       1 -19 1077m 380m  18m S    4  9.9   1:37.46 snort
16837 root       1 -19 1077m 379m  18m S    4  9.9   1:37.72 snort
 5052 root      20   0 1737m 4508 3784 S    2  0.1 138:40.09 adi
 5091 root      10 -10  266m 2660 2284 S    2  0.1 133:05.54 ndmain.bin
 6840 root      20   0  555m 9020 3136 S    2  0.2 240:05.88 sftunnel
 6844 root      20   0  309m 116m 6872 S    2  3.1 282:50.72 run_hm.pl
 8640 admin     20   0 15240 1160  852 R    2  0.0   0:00.02 top
    1 root      20   0  4232  660  628 S    0  0.0   0:25.05 init
    2 root      20   0     0    0    0 S    0  0.0   0:00.58 kthreadd
    3 root      20   0     0    0    0 S    0  0.0   0:04.41 ksoftirqd/0
    4 root      20   0     0    0    0 D    0  0.0   0:00.10 kworker/0:0
    5 root       0 -20     0    0    0 S    0  0.0   0:00.00 kworker/0:0H
    7 root      RT   0     0    0    0 S    0  0.0   0:00.84 migration/0
    8 root      20   0     0    0    0 S    0  0.0   0:00.00 rcu_bh
    9 root      20   0     0    0    0 S    0  0.0  19:01.74 rcu_sched
```

Figure 14 Top command output from Cisco ASA Firepower

5.2 Sophos UTM

Sophos UTM firewall is a security solution from Sophos including all required capabilities for a next generation firewall. Sophos UTM also extends the protection into email and web application areas. Sophos UTM houses Snort IPS solution.

Sophos UTM with software version 9.505-4 includes Snort version 2.9.9.0. The total of XX signatures were loaded into the Snort engine in the test setup. Hardware wise UTM model 110 has Intel Atom CPU N450 with clock rate at 1.66GHz and 8 gigabytes of RAM. Sophos datasheet promises IPS throughput up to 140 Mbps (Sophos 2012).

Figure 15 presents the results from the first test. A total of 210 attacks were executed where 180 of them were allowed and 30 were blocked by the Sophos UTM.

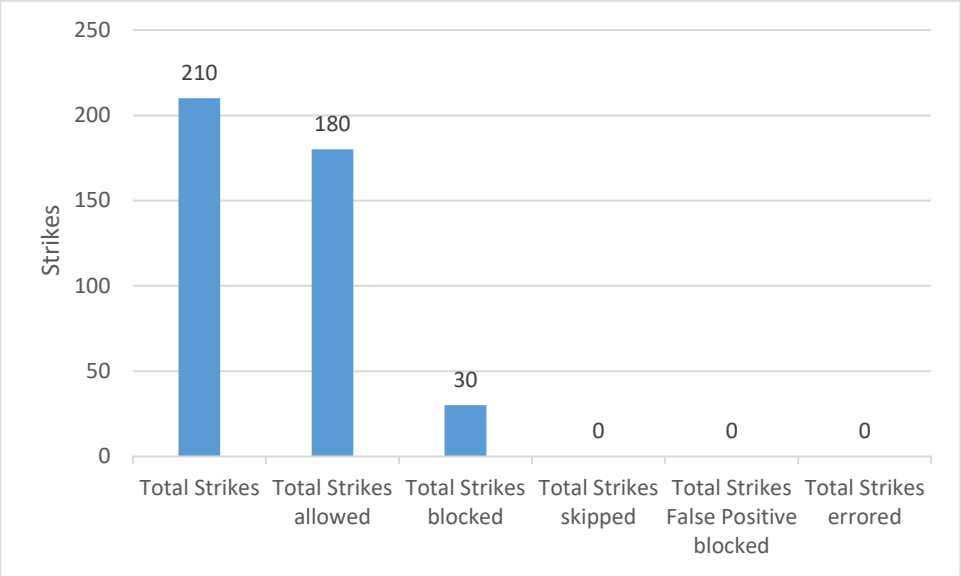


Figure 15 Sophos UTM first test detection rate

In total Sophos UTM was able to block 14.29% of the executed attacks and 85.71% of the attacks were allowed during the first test run. Appendix 4 shows the detailed strike category assessment of the first test. Detection rate can be considered quite low as the Strike Level 1 – 2017 attack pack includes strikes from year 2017 with CVSS score 10.0. This is caused by the fact mentioned by Hogue & Carter (2006) that "*if your IPS does not efficiently manage resources when maintaining state, then the large consumption of resources (such as memory and CPU) can lead to a slow response time, dropped packets, missed signatures, and so on, which adversely impacts the effectiveness of your IPS.*" As the number of required signatures is too high for the available hardware to handle the IPS manufactures have to reduce the amount of signature loaded into the memory which then leads to reduced detection rates in low end devices.

In the second test, Sophos UTM was able to block 28% of the executed attack and 72% of the attacks were allowed during the test run. One of the attacks failed due to an error as seen in Figure 16. Appendix 5 includes the detailed strike category assessments.

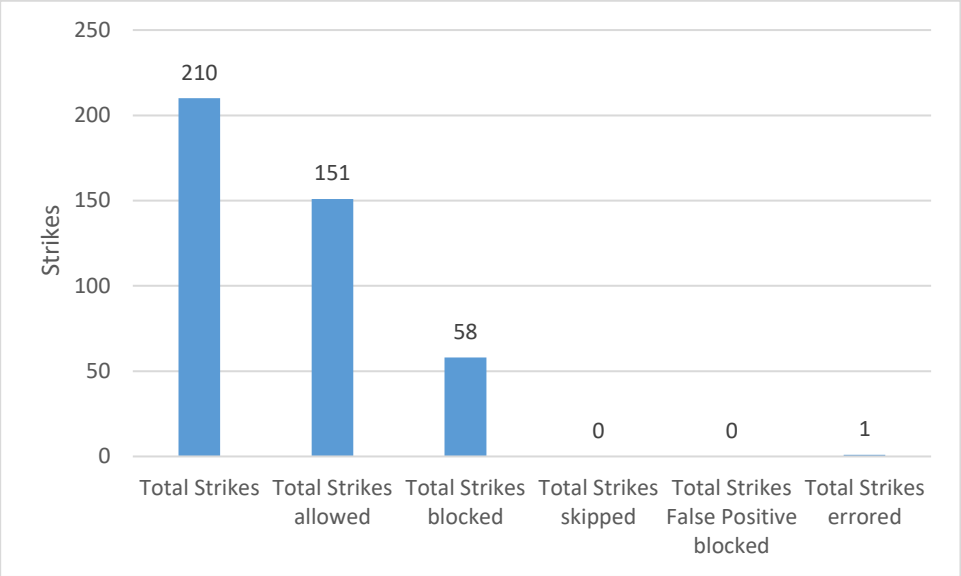


Figure 16 Sophos UTM second test detection rate

The average throughput of the device was around 50Mb/s and the throughput stayed solid throughout the test run. When compared to the figures promised in Sophos UTM datasheet with IPS throughput of 140Mb/s, it can be concluded that IPS core traffic profile of the Ixia BreakingPoint causes an extreme load to the IPS systems. Figure 17 presents the throughput of the Sophos UTM throughout the second test run.

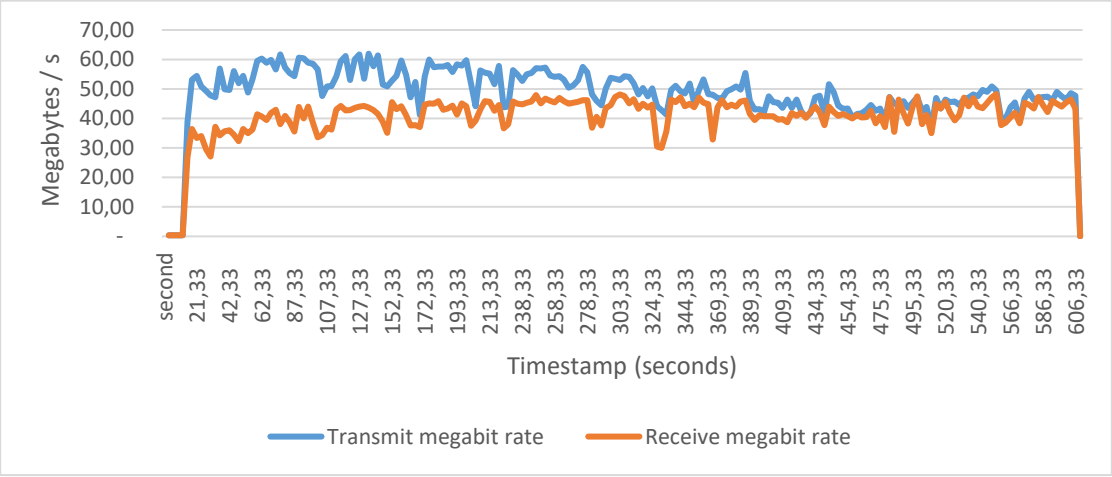


Figure 17 Sophos UTM second test throughput

Table 3 presents the detailed frame and application transaction analysis of the second test. From Table 3 it can be noted that the 12 percent of the application transaction failed due to the TCP retry limit, which indicates that the test profile combined with the test environment was causing too many TCP connections through the Sophos UTM. Additionally, quite many transactions failed because of the resolve

receive timeout, which implies that the UTM was not able to pass all of the DNS requests from the virtual client side into the virtual server side. Appendix 7 displays the Snort process CPU load throughout the second test.

Table 3 Sophos UTM second test frame analysis

Measurement	Value
Frames transmitted	4 593 365
Frames received	4 088 960
Frame data transmitted	3 704 366
Frame data received	3 067 328
Attempted	8277
Aborted	0
Successes	5474
Failures due to external events	187
Failures due to ramp down	933
Failures due to TCP retry limit	1019
Failures due to UDP receive timeout	61
Failures due to resolve receive timeout	787

In the third test, Sophos UTM was able to block 14% of the executed attacks and 86% of the attacks were allowed during the test run. Appendix 8 shows the detailed strike category assessment of the third test. Figure 18 presents the detection rate of the third test run.

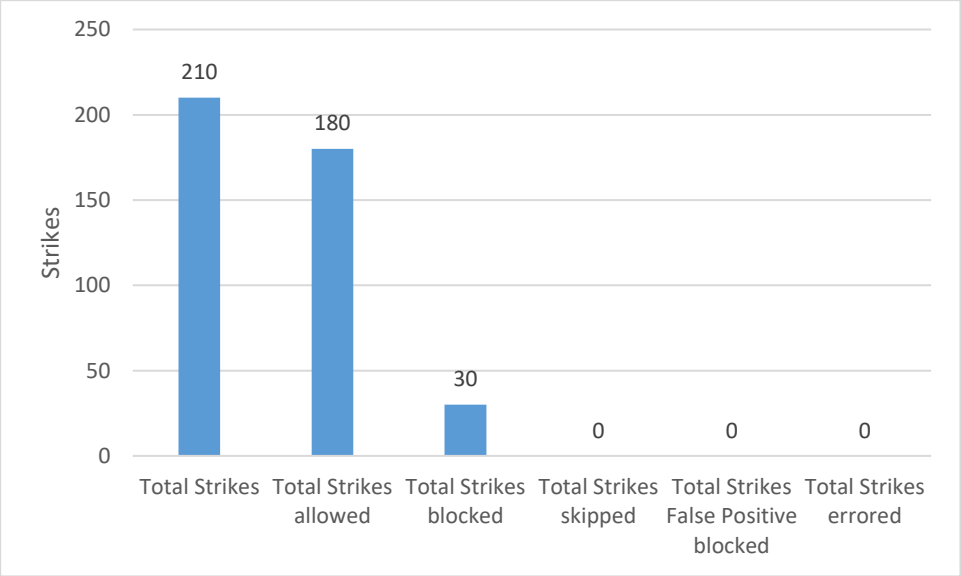


Figure 18 Sophos UTM third test detection rate

The throughput of the third test was slightly lower, around 30Mb/s when compared to the 50Mb/s in the second test run. The result is somewhat surprising due to the

fact that the NGFW Enterprise Perimeter Traffic Mix 2016 used in the third test should be much lighter for the IPS systems than the IPS Core Traffic pattern used in the second test. Figure 19 presents the throughput of the Sophos UTM during the third test.

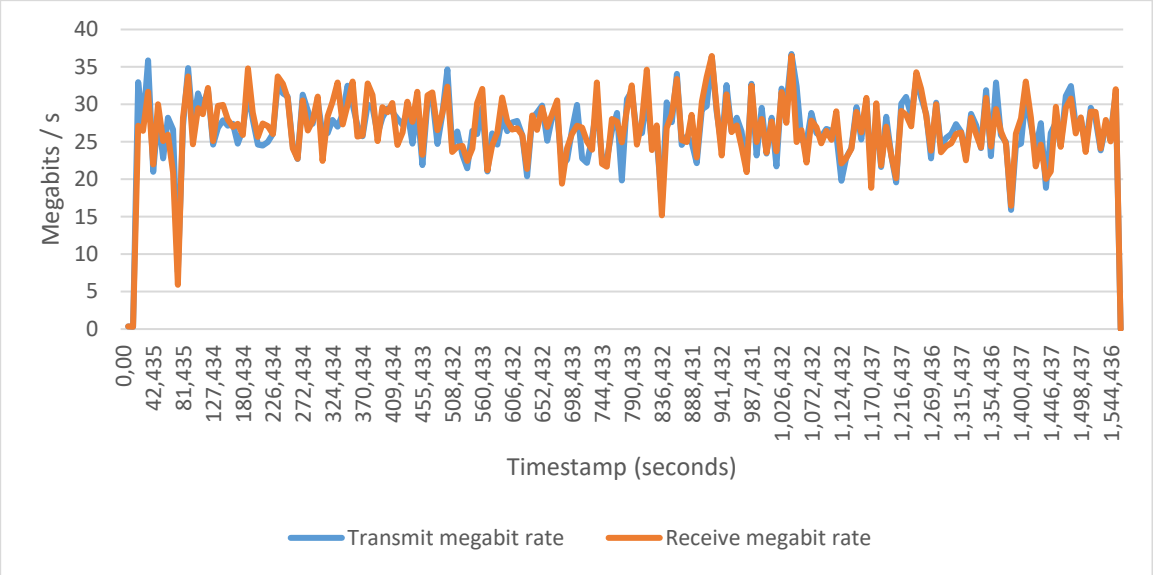


Figure 19 Sophos UTM third test throughput

The frame and application transaction analysis in Table 4 shows that no major failures occured during the third test. For example, when comparing the failures due to the TCP retry limit in to the second test, the rate is only 0.23%. Appendix 9 displays the application transaction summary from the third test. The Snort process CPU load remained extremely high throughout the third test which is presented in Appendix 10.

Table 4 Sophos UTM third test frame analysis

Measurement	Value
Frames transmitted	8 444 524
Frames received	8 386 884
Frame data transmitted	4 991 969 277
Frame data received	4 974 163 265
Attempted	110 477
Aborted	0
Successes	109 328
Failures due to external events	279
Failures due to ramp down	870
Failures due to TCP retry limit	10
Failures due to UDP receive timeout	7

Failures due to resolve receive timeout	250
Failures due to a premature session close	0
Failures due to a premature Super Flow close	12

For the fourth test, a CVE-2017-5689 vulnerability where an unprivileged network attacker could gain system privileges to provisioned Intel manageability SKUs: Intel Active Management Technology (AMT) and Intel Standard Manageability (ISM) was run ten times against Sophos UTM. Figure 20 presents the result where it can be seen that the Sophos UTM was able to prevent all ten attacks.

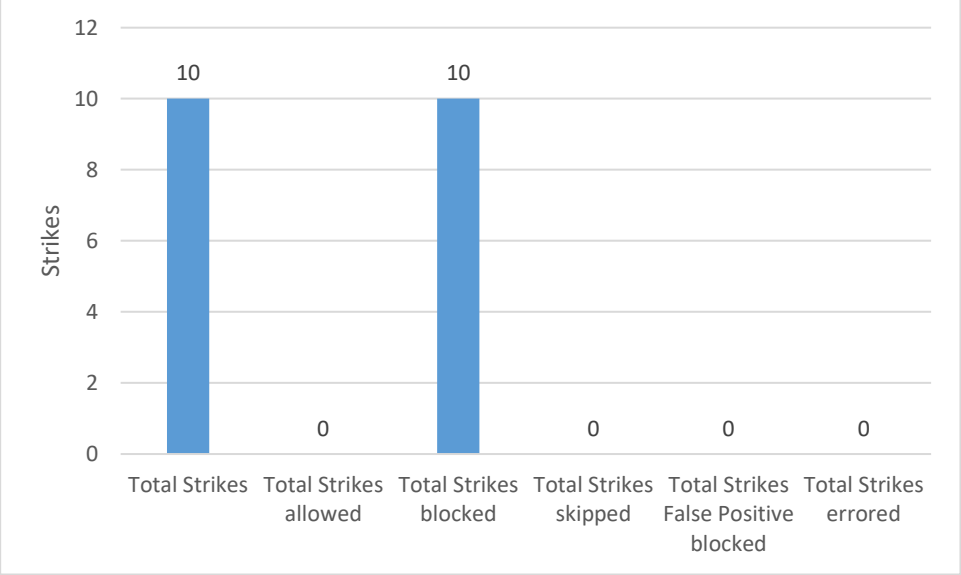


Figure 20 Sophos UTM fourth test detection rate

5.3 Cisco ASA Firepower 5506

Cisco Asa Firepower is a new next generation firewall platform from Cisco. ASA5506-X with FirePOWER Services combines Cisco's proven network firewall with the industry's most effective next-gen IPS and advanced malware protection (Cisco). ASA Firepower 5506 includes Snort IPS solution.

Cisco ASA Firepower 5506 with software version 6.2.0.3 has snort version 2.9.11. A total of XX rules were loaded for the test scenario. Firewall has Atom C2000 series 1250 MHz CPU with 4 cores and 4Gb of RAM. Cisco ASA Firepower 5506 datasheet shows combined AVC and IPS throughput to be 125 Mbps (Cisco).

Figure 21 presents results from the first test. A total of 210 attacks were executed where 171 of them were allowed and 39 were blocked by the Cisco Asa Firepower.

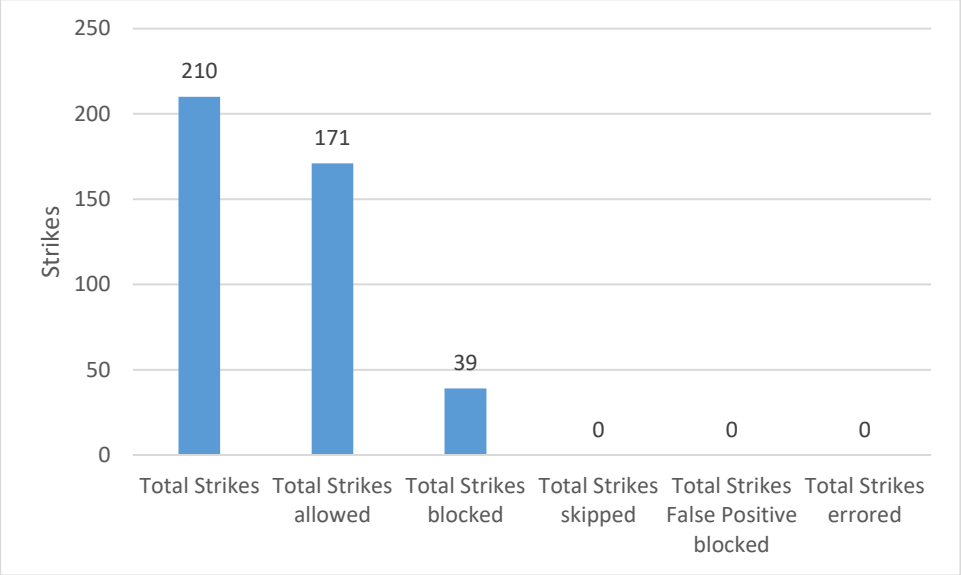


Figure 21 Cisco Asa Firepower first test detection rate

In total, Cisco Asa Firepower was able to block 18.57% of the executed attacks and 81.4% of the attacks were allowed during the first test run. Appendix 11 shows the detailed strike category assessment of the first test. Low percentage in detection rate of the Cisco Asa Firepower shows that the amount of signatures is highly optimized also in Cisco product.

In the second test, Cisco Asa Firepower was able to block 28% of the executed attacks and 72% of the attacks were allowed during the test run as seen in Figure 22. Appendix 12 includes the detailed strike category assessments.

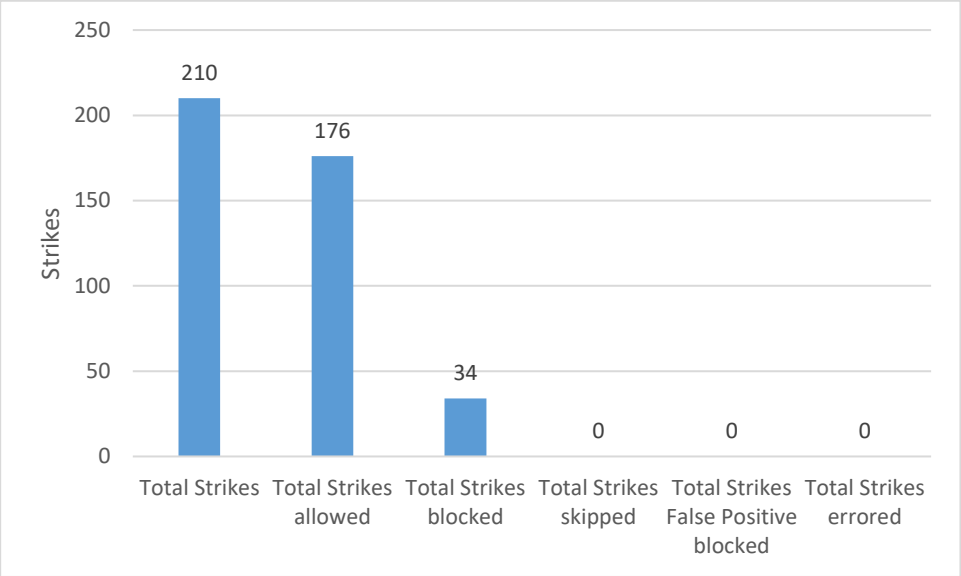


Figure 22 Cisco Asa Firepower second test detection rate

Figure 23 shows that the average throughput of the Cisco Asa Firepower was around 50Mb/s and varied slightly throughout the test run. Cisco Asa Firepower 5506 datasheet promises that the IPS throughput should be around 125Mb/s.

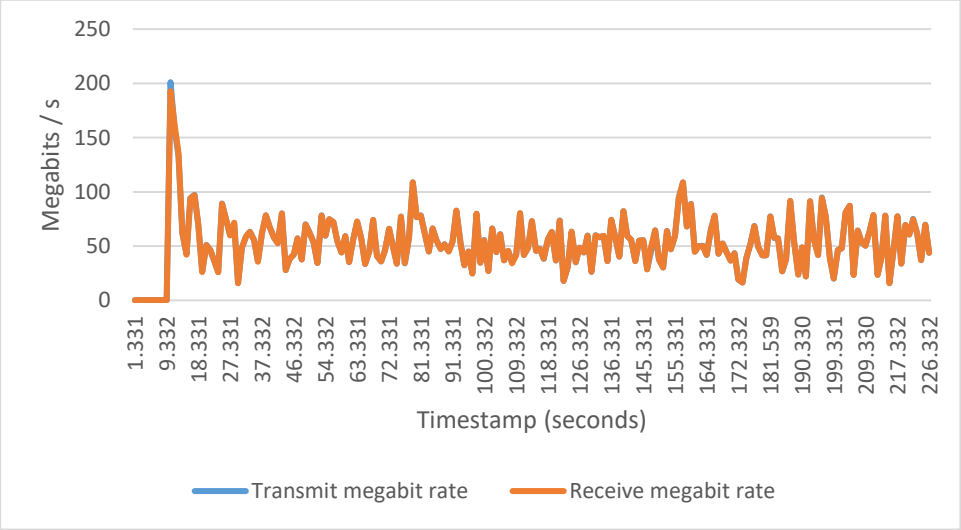


Figure 23 Cisco Asa Firepower second test throughput

Table 5 presents the detailed frame and application transaction analysis of the second test. Statistics in Table 5 shows that the major part of the failed application transactions during the second test was caused by the resolve receive timeout. For some reason, Cisco Asa Firepower was not able to pass the DNS requests from the virtual client side into the virtual server side of the test environment.

Table 5 Cisco Asa Firepower second test frame analysis

Measurement	Value
Frames transmitted	573 215
Frames received	467 745
Frame data transmitted	83 165 881
Frame data received	58 245 391
Attempted	84 090
Aborted	0
Successes	30 555
Failures due to external events	53 081
Failures due to ramp down	454
Failures due to TCP retry limit	12 666
Failures due to UDP receive timeout	0
Failures due to resolve receive timeout	40 415

In total, 63% of the application transactions failed in the second test and 76% of the failures were caused by the resolve receive timeout behavior. Appendix 13 presents the summary of the second test application transactions.

In the third test, Cisco Asa Firepower was able to block 28% of the executed attacks and 86% of the attacks were allowed during the test run. Appendix 14 shows the detailed strike category assessment of the third test. Figure 24 presents the detection rate of the third test run.

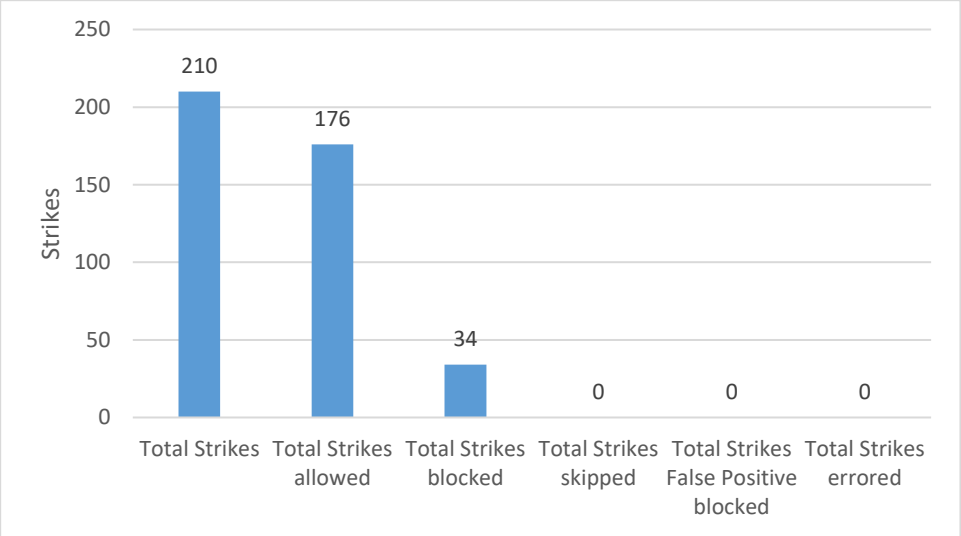


Figure 24 Cisco Asa Firepower third test detection rate

The throughput shown in Figure 25 remains the same as in the second test, around 50Mb/s. This is a major difference compared to the behavior of the Sophos UTM where the throughput between tests 2 and 3 had a difference.

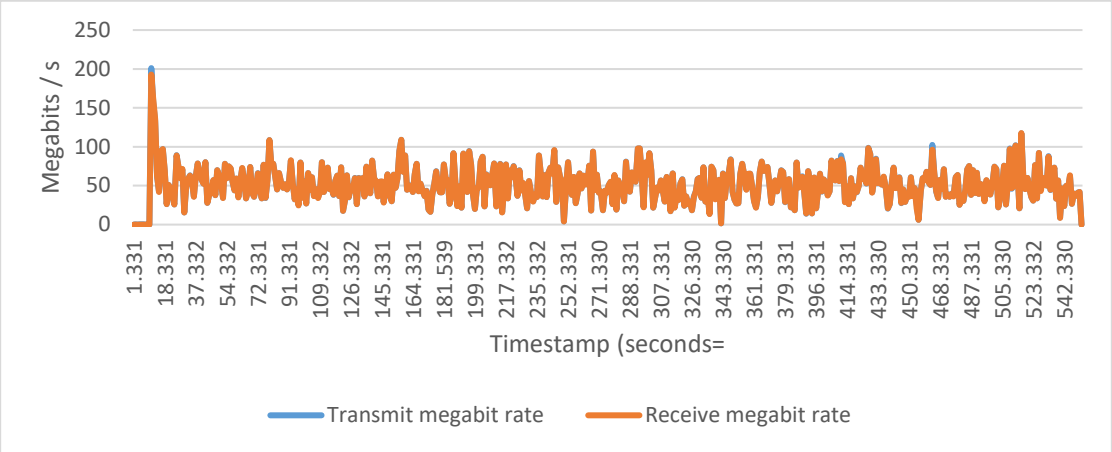


Figure 25 Cisco Asa Firepower third test throughput

The frame and application transaction analysis in Table 6 shows that the same behavior continues where the major part of the application transactions is failing due to the resolve receive timeout. Appendix 15 presents the summary of the third test application transactions.

Table 6 Cisco Asa Firepower third test frame analysis

Measurement	Value
Frames transmitted	6 841 219
Frames received	6 805 521
Frame data transmitted	3 375 612 266
Frame data received	3 371 536 495
Attempted	62 662
Aborted	0
Successes	27 888
Failures due to external events	34 468
Failures due to ramp down	306
Failures due to TCP retry limit	7
Failures due to UDP receive timeout	0
Failures due to resolve receive timeout	34 461

In the fourth test, a vulnerability MS17-010 where remote code execution vulnerabilities exist in the way that the Microsoft Server Message Block 1.0 (SMBv1) server handles certain requests was launched against Cisco Asa Firepower. An attacker who successfully exploited the vulnerabilities could gain the ability to execute code on the target server. Figure 26 presents the detection rate for test four.

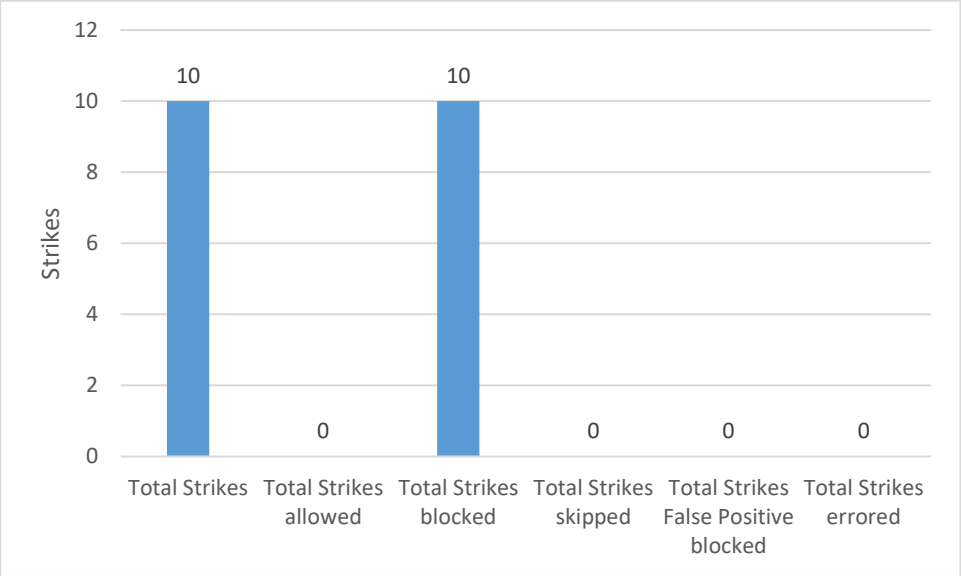


Figure 26 Cisco Asa Firepower fourth test detection rate

5.4 OPNsense

OPNsense is an open source FreeBSD based firewall and routing platform. OPNsense includes most of the features available in expensive commercial firewalls. OPNsense started as a fork of pfSense® and m0n0wall in 2014, with its first official release in January 2015. (OPNSense) OPNsense is equipped with Suricata IPS solution for threat detection and analysis.

OPNsense version OPNsense 17.7.7_1 includes Suricata version 4.0.1. A total of XX rules were loaded into the memory for the test scenarios. OPNsense is presented as a virtual firewall running in VMWare hypervisor where 1 CPU core and 8Gb of memory were allocated for the guest system.

Figure 27 presents the results from the first test. Total of 210 attacks were executed with 169 of them allowed and 41 blocked by the OPNsense.

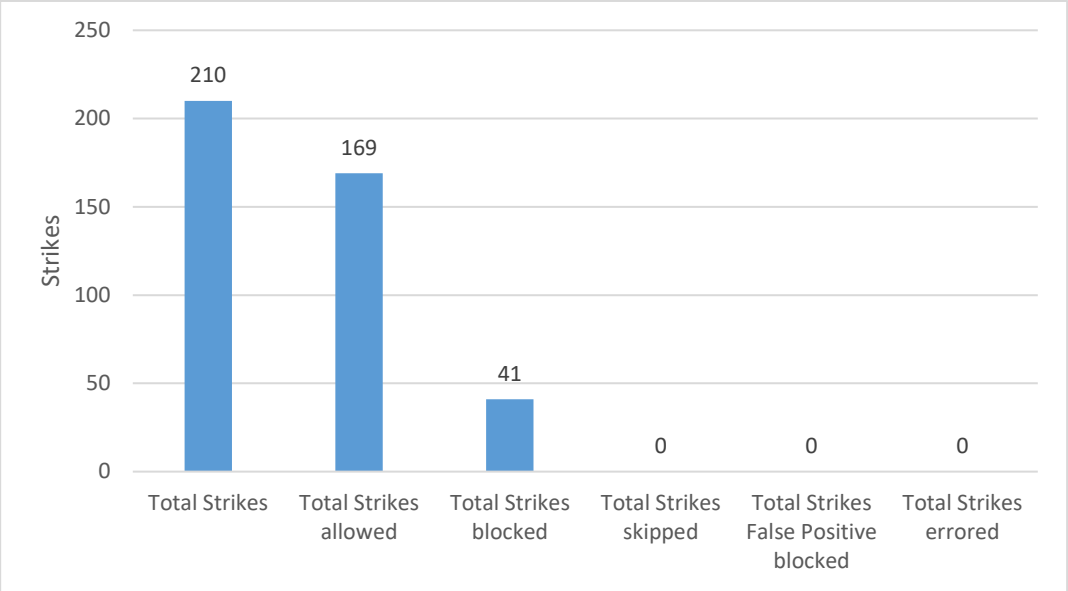


Figure 27 OPNsense first test detection rate

In total, OPNsense was able to block 19.5% of the executed attacks and 80.5% of the attacks were allowed during the first test run. Appendix 16 shows the detailed strike category assessment of the first test. Also in OPNsense case the amount of IPS signatures has to be reduced in order to avoid heavy performance drop.

Figure 28 shows that in the second test, OPNsense was able to block 48.5% of the executed attack and 51.5% of the attacks were allowed. Appendix 17 includes the detailed strike category assessments.

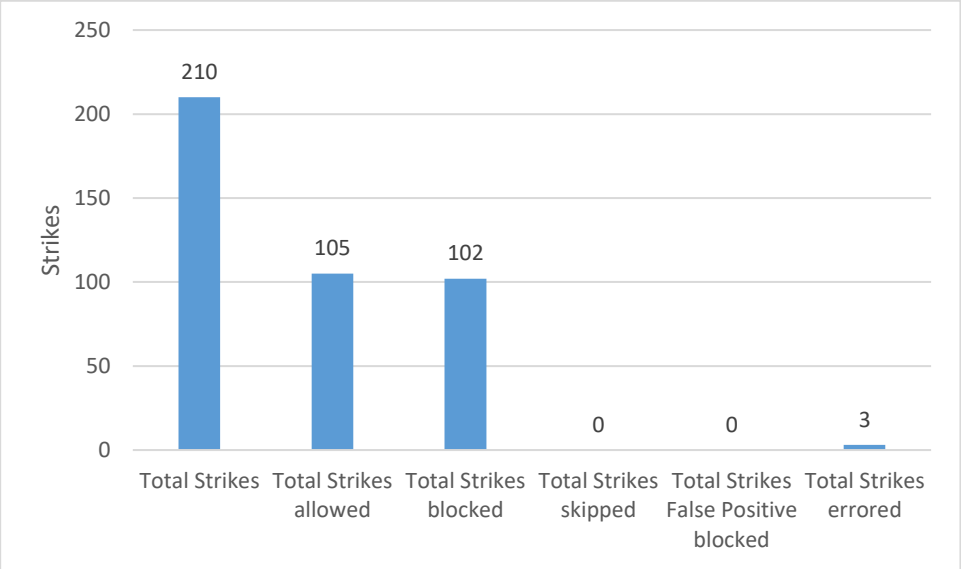


Figure 28 OPNsense second test detection rate

As illustrated in Figure 29, a major difference can be seen between the transmitted and received megabit rate in the second test run.

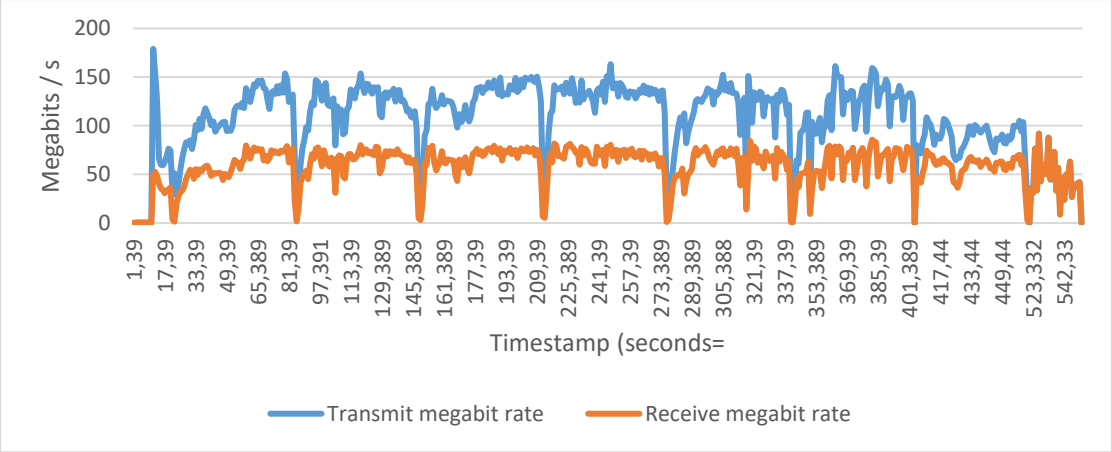


Figure 29 OPNsense second test throughput

Table 7 confirms the findings presented in Figure 29. Over half of the transmitted frames are not received by the BreakingPpoint tool through the virtual test networks. According to the Table 7, application transactions are failing with multiple clauses, which is a significant difference when compared to the Cisco Asa Firepower tests where a single cause of the transaction failures was detected. In the application transactions summary in Appendix 18, it can be seen that OPNsense struggles with HTTP and DNS traffic. Suricata process CPU load remained high throughout the second test as seen in Appendix 19.

Table 7 OPNsense second test frame analysis

Measurement	Value
Frames transmitted	6 470 367
Frames received	2 893 187
Frame data transmitted	5 677 871 194
Frame data received	2 985 615 717
Attempted	82 347
Aborted	0
Successes	53 716
Failures due to external events	28 154
Failures due to ramp down	477
Failures due to TCP retry limit	15 343
Failures due to UDP receive timeout	100
Failures due to resolve receive timeout	12 603
Failures due to a premature session close	0
Failures due to a premature Super Flow close	99
General application failures	9

Figure 30 shows the detection rate in the third test run. OPNsense was able to block 41% of the attacks and allowed 59% of the attacks. The difference between the second and third test in blocked attacs is 15.5%. Appendix 20 includes the detailed strike category assessments.

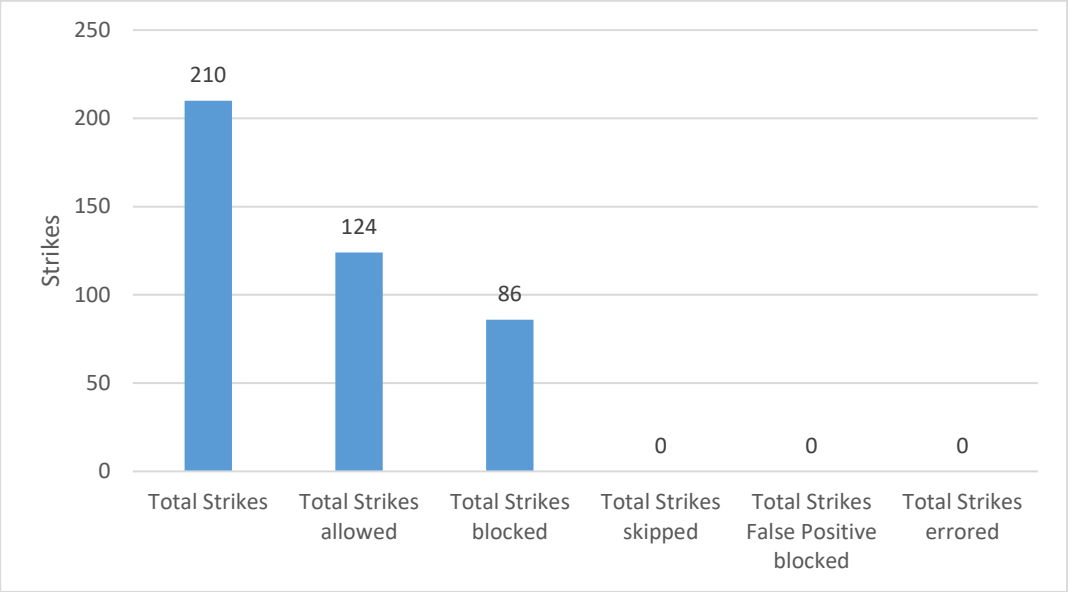


Figure 30 OPNsense third test detection rate

Figure 31 and Table 8 confirm that similar behavior where there is a major difference between transmitted and received frames continues also in test run 3. Again, diverse causes of failures are seen in application transactions failures.

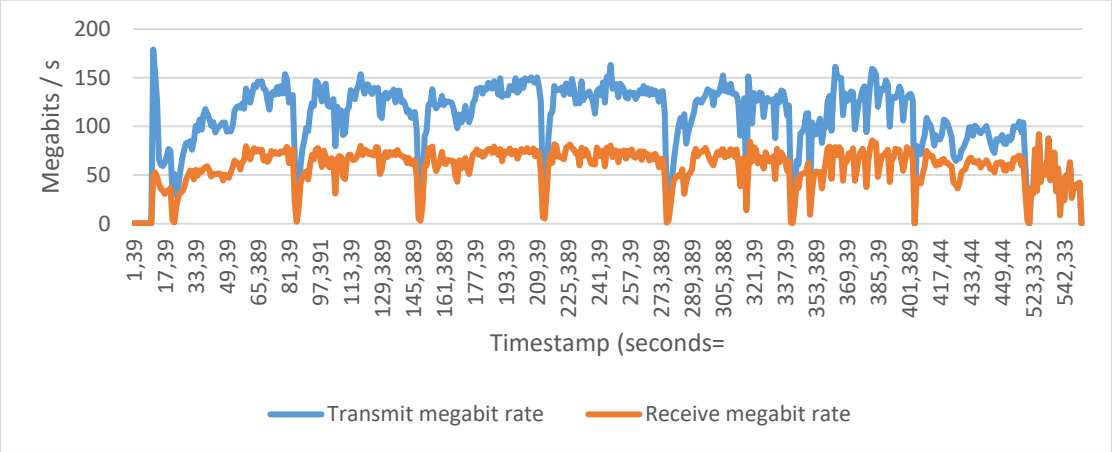


Figure 31 OPNsense third test throughput

Appendix 21 shows that most application transaction failures come from HTTP, HTTPS and DNS traffic. CPU load of the Suricata process seen in Appendix 22 shows similar behavior when compared to the second test.

Table 8 OPNsense third test frame analysis

Measurement	Value
Frames transmitted	5 467 345
Frames received	2 955 998
Frame data transmitted	3 954 431 188
Frame data received	2 563 138 737

Attempted	81 058
Aborted	0
Successes	54 336
Failures due to external events	26 714
Failures due to ramp down	530
Failures due to TCP retry limit	11 982
Failures due to UDP receive timeout	34
Failures due to resolve receive timeout	12 285
Failures due to a premature session close	0
Failures due to a premature Super Flow close	2413

5.5 Conclusion

Initial detection rates provided by the first test showed that each of the tested systems was able to detect an almost similar amount of malicious traffic where Sophos UTM blocked 30 attacks, Cisco ASA Firepower blocked 39 and OPNsense was able to block 41 attacks. These results are a good example of the problem in high speed intrusion detection where the number of signatures must remain low. Otherwise the performance requirements of the hardware platform would not be cost effective. Panthan (2014) also addresses this problem saying that "*Dedicated hardware accelerators become a necessity to address these challenges.*" However especially in low end devices such option is not often available.

The difference in the detection rates increased when IPS Core and NGFW traffic patterns were added into the test scenario. Also, the way with which the devices under the test handled the background traffic provided by the patterns showed some major differences.

Sophos UTM was proven to be the most solid contender when comparing the handling of the background traffic in a high load situation. When looking at the frame analysis in Table 3, no single cause of the application transaction failure can be distinguished. With IPS Core traffic pattern, the detection rate rose to 58; however, with NGFW traffic pattern the detection rate decreased back to 30 attacks. The better detection rate in test number two is probably caused by the device dropping the traffic before inspecting as with the more lighter traffic profile in test three, the detection rate lowered back to the baseline.

Cisco ASA Firepower showed some strange behavior with traffic patterns as a major part of the application transactions were failing because of the DNS resolve timeout as seen in Tables 5 and 6. An error message “interface br0 lost contact” was observed during the tests two and three in the management console of the devices, which leads to the suspicion that some other processing or component before the actual Snort process was limiting the throughput of the traffic. This assumption was even more confirmed when the manufacturer community forums were able to prove that the ASA Firepower 5506 has two modules, plain firewall side and IPS side connected together by an internal bridge interface (Cisco Community). Therefore, Cisco ASA 5506 tests cannot be considered a success.

OPNsense was a leading contender in test 1 with 41 blocked attacks. When adding the background traffic in tests 2 and 3, the detection rates increased even more. As it can be seen from the data provided by the tests 2 and 3, OPNSense had major problems when it comes to the handling of the traffic passing through the device. Figure 29 of test 2 and Figure 31 of test 3 show that over half of the traffic was either not passed through the device or erred in some way during the transit. Especially the handling of the HTTP based streaming content was proven to be a major problem for the Suricata IPS in OPNsense platform. Since over half of the traffic was dropped by the device, the higher detection rates cannot be considered to be valid and they were probably just randomly dropped.

In the case of the Sophos UTM and OPNsense, the CPU load of the IPS process was successfully raised near the 100% throughout the tests. With Cisco Asa Firepower, an internal bridge interface connecting the IPS and firewall side was limiting the traffic reaching the IPS process. This led to the situation where the IPS load in Cisco Asa Firepower was not high enough to provide a fair comparison with other tested systems.

Overall the results of the test scenarios cannot be considered accurate enough for a proper conclusion about the IPS detection rates in high load scenario. Only Sophos UTM test results reach such confidence level where a proper conclusion level that the detection rate of the snort IPS solution built in the device is not affected by the extremely high system load. Other devices under the test had major issues with handling of the background traffic, which rendered the results unusable.

6 Discussion

The goal of the research was to study if a high system load of an IPS system affects the detection rate of the malicious traffic passing through the system. Additionally, the research was thought to present a scenario where a possible attacker could benefit from the high load scenario caused by, for example, a denial of service attack.

The devices under the research were chosen based on the IPS solution and not by the manufacturer. One open source product was chosen to provide a comparison between commercial and free solutions. As most of the commercial products rely on a Snort solution for IPS functionality, the open source product OPNsense was chosen as it relies on Suricata IPS solution. The device models were deliberately selected from the low end of the product catalogs so that the high load scenario could be easily achieved.

Ixia BreakingPoint tool used to test the devices proved to be a comprehensive tool for such tests as it was able to provide the background traffic as well as malicious traffic. Ixia-made traffic patterns and attack patterns were chosen to keep the test setup simple enough as the Ixia tool has quite many options to tweak, for example the amount or type of the malicious traffic has over 20 options.

The first problem which arose during the testing was the diversity of the background network traffic. For example, the traffic could be using a high amount of connections with relatively low bandwidth or vice versa. This led to the problem of sizing the correct amount of traffic so that the device would experience a high load but in the meantime, keep the connection rate low enough so that the device would not run out of memory. Ixia calls the point where the device under the test is not able to handle the traffic a breaking point. The researcher believes that with more accurate breaking point sizing, the results would be more accurate.

The second problem observed during the tests was the amount of IPS signatures loaded into each of the devices. The more loaded signatures, the easier it was to saturate the IPS process in the device. A decision was made to keep the number of signatures on default setting in each of the device. This led to an uneven amount of signatures between the tested devices.

The third major problem observed during testing was the difference in results between the BreakingPoint tool and the tested system. In many cases, the IPS system would report the malicious traffic as blocked; however, the BreakingPoint tool reported a successful attack. A comprehensive comparison between the test tool reporting and IPS system reporting would have probably fined the detection rate results in the research.

Overall, the research produced quite mixed results. Only one of the tested solutions, Sophos UTM was able to give solid results as Cisco ASA Firepower failed to deliver the traffic into Snort process and OPNsense's Suricata was unable to handle HTTP based streams. Additionally, the complexity of the background traffic patterns and IPS signatures caused the test scenario to easily become too complex to handle in the scope of the research.

Further analysis should concentrate on providing a more accurate breaking point of the tested devices. Also, a more detailed look should be taken into the background traffic profiles and IPS signature patters loaded into the memory of the devices.

References

About OPNsense. Web page on OPNsense website. Accessed on 15 December 2017. Retrieved from https://opnsense.org/about/about-opnsense/

Adesina O.,Barker K., Burns D. 2012. CCNP Security IPS 642-627 Official Cert Guide Chapter 1. Intrusion Prevention and Intrusion Detection Systems. Cisco Press.

Adesina O.,Barker K., Burns D. 2012. CCNP Security IPS 642-627 Official Cert Guide Chapter 3. Network IPS Traffic Analysis Methods, Evasion Possibilities, and Anti-evasive Countermeasures. Cisco Press.

Aruba Networks Wireless intrusion detection. 2014. PDF document on Aruba networks website. Accessed on 10 October 2017. Retrieved from http://www.arubanetworks.com/assets/tg/TG_WIP.pdf

ASA and Firepower hardware fact sheet. Page on Cisco Systems community web site. Accessed on 23 December 2017. Retrieved from https://communities.cisco.com/community/technology/security/ngfw-firewalls/blog/2016/02/02/asa-hardware-facts-sheet

Barker K., Morris S. 2012. CCNA Security 640-554 Official Cert Guide Chapter 15. Cisco IPS_IDS Fundamentals. Cisco Press.

Cisco ASA with FirePOWER Services Data Sheet. Page on Cisco Systems web site. Accessed on 23 December 2017. Retrieved from https://cisco-apps.cisco.com/c/en/us/products/collateral/security/asa-5500-series-next-generation-firewalls/datasheet-c78-733916.html

Cisco ASA 5500-X with FirePOWER Services. Page on Cisco Systems web site. Accessed on 23 December 2017. Retrieved from https://www.cisco.com/c/en/us/products/security/asa-firepower-services

Hogue J., E Carter. 2006. Intrusion Prevention Fundamentals Chapter 1. Intrusion Prevention Overview. Cisco Press

Ixia Breakingpoint Ve data sheet. Page on Ixia web site. Accessed on 15 November. Retrieved from https://www.ixiacom.com/resources/breakingpoint-virtual-edition-ve

National vulnerability database. CVE-2017-5689 detail. Accessed on 5 December 2017. Retrieved from https://nvd.nist.gov/vuln/detail/CVE-2017-5689

NSS Labs Test Methodology – Next Generation Firewall v7.0. Page on NSS labs web site. Accessed on 5 October 2017. Retrieved from https://www.nsslabs.com/linkservid/CC75A111-5056-9046-93B6183362701160/

NSS Labs Test Methodology – Next Generation Intrusion Prevention System (NGIPS) v3.1. Page on NSS labs web site. Accessed on 5 October 2017. Retrieved from https://research.nsslabs.com/reportaction/report-500/Toc?SearchTerms=ngips

Panthan A. 2014. The State of the Art in Intrusion Prevention and Detection. Taylor & Francis Group, LLC

Trost R. 2009. Practical Intrusion Analysis: Prevention and Detection for the Twenty-First Century. Addison-Wesley Professional.

Shuttleworth, M. 2008. Case Study Research Design. Article on Ex-plorable.com web page. Accessed on August 2017. Retrieved from https://explorable.com/case-study-research-design.

SNORT Users Manual 2.9.9. November 14, 2016. Page on snort.org web page. Accessed on 4.November. Retrieved from http://manual-snort-org.s3-website-us-east-1.amazonaws.com/

Sophos Intercept X. Picture on Sophos web page. Accessed on 1. October 2017. Retrieved from www.sophos.com

Sophos UTM 110/120 datasheet. PDF document on Sophos web page. Accessed on 25 November. Retrieved from https://www.sophos.com/enus/medialibrary/pdfs/factsheets/sophosutm100110120dsna.pdf

Suricata User Guide. 2016. Web page on Suricata web site. Accessed on 25 November. Retrieved from http://suricata.readthedocs.io/en/latest/

Suricata IDS features. Web page on Suricata web site. Accessed on 25 November. Retrieved from www.suricata-ids.org/features/

What is an intrusion detection system. 2017. Page on Palo Alto Networks cyberpedia. Accessed on January 2018. Retrieved from https://www.paloaltonetworks.com/cyberpedia/what-is-an-intrusion-detection-system-ids

What is an intrusion prevention system. 2017. Page on Palo Alto Networks cyberpedia. Accessed on January 2018. Retrieved from https://www.paloaltonetworks.com/cyberpedia/what-is-an-intrusion-prevention-system-ips

Appendice

Appendix 1. Load profile values for Ramp Up Behavior

Parameter	Description	Valid values	Description
Ramp Up Behavior	Sets how the component will open sessions during the ramp up phase.	Full Open	The full TCP handshake performed when sessions are opened.
		Full Open + Data	The full TCP handshake performed when sessions are opened. Data sent once the session opens.
		Full Open + Data + Full Close	The full TCP handshake performed when sessions are opened. Data sent once the session opens. Sessions are closed as they finish sending data and new sessions are opened in their place.
		Full Open + Data + Close with Reset	The full TCP handshake performed when sessions are opened. Data sent once the session opens. A TCP close with a RST is initiated and the TCP close state machine is bypassed.
		Half Open	The full TCP handshake performed when sessions are opened, but the final ACK is omitted.
		SYN Only	Only SYN packets are sent.
		Data Only	Only PSH data packets are sent. The state machine is bypassed, so no connections are set up; therefore, the ACKs will be invalid.

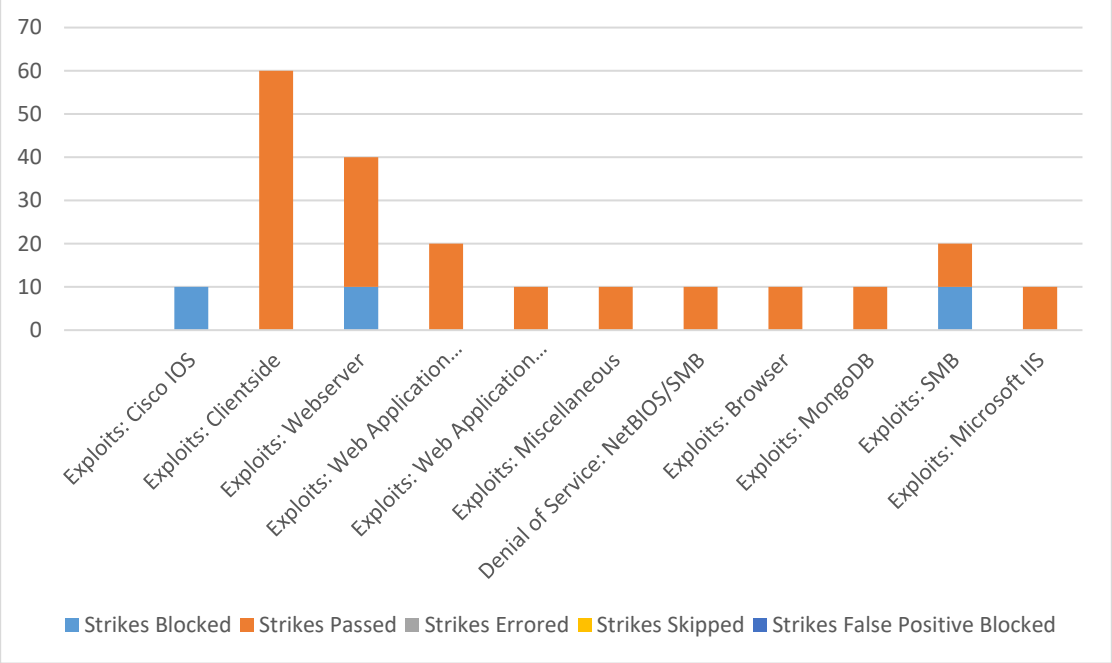
Appendix 2. Load profile values for Steady-State Behavior

Parameter	Description	Valid values	Description
Steady-State Behavior	Sets how the component will handle sessions during the steady-state phase.	Open and Close Sessions	Sessions are closed as they finish sending data, and new sessions are opened.
		Hold Sessions Open	No existing sessions opened during Ramp Up are closed.
		Open and Close with Reset	A TCP close with a RST is initiated and the TCP close state machine is bypassed
		Open and Close with Reset Response	Once a session is closed, the server will respond with a RST and change to the TCP CLOSED state. This option bypasses the TCP TIME_WAIT state.

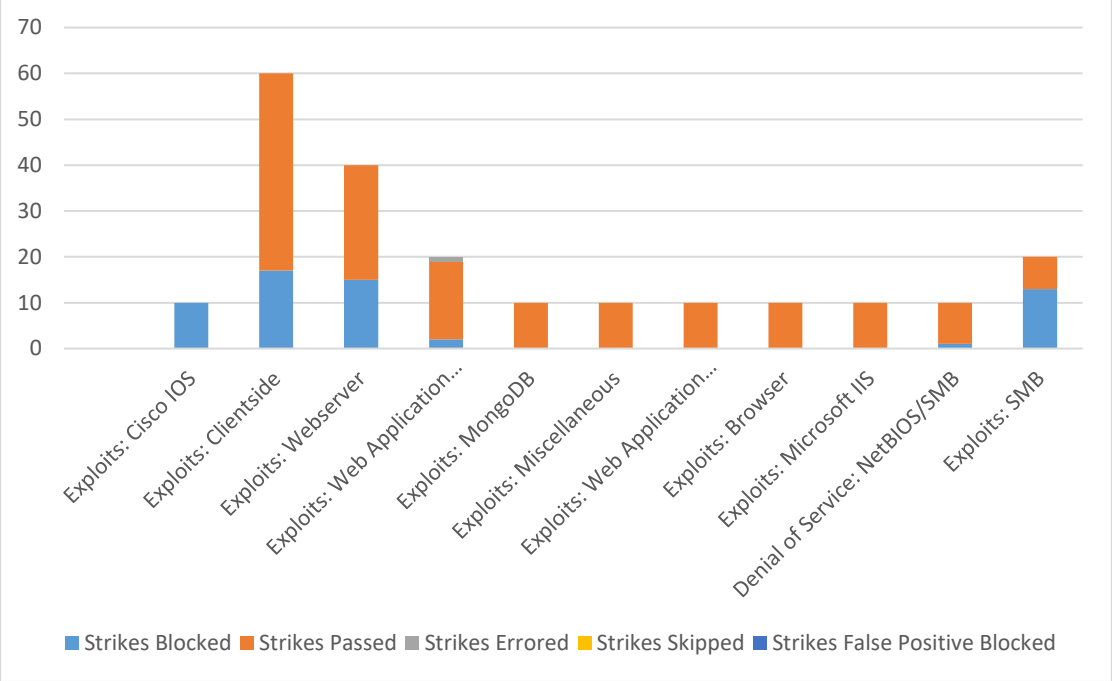
Appendix 3. Load profile values for Ramp Down Behavor

Parameter	Description	Valid values	Description
Ramp Down Behavior	Sets how the component will close sessions during the time period specified for Ramp Down Duration	Full Close	The full TCP session close is performed.
		Half Close	The full TCP session close is performed, but the final ACK is omitted
		Reset	Close all sessions by sending TCP RST (reset) packets.
		Reset Response	Open and Close with Reset Response

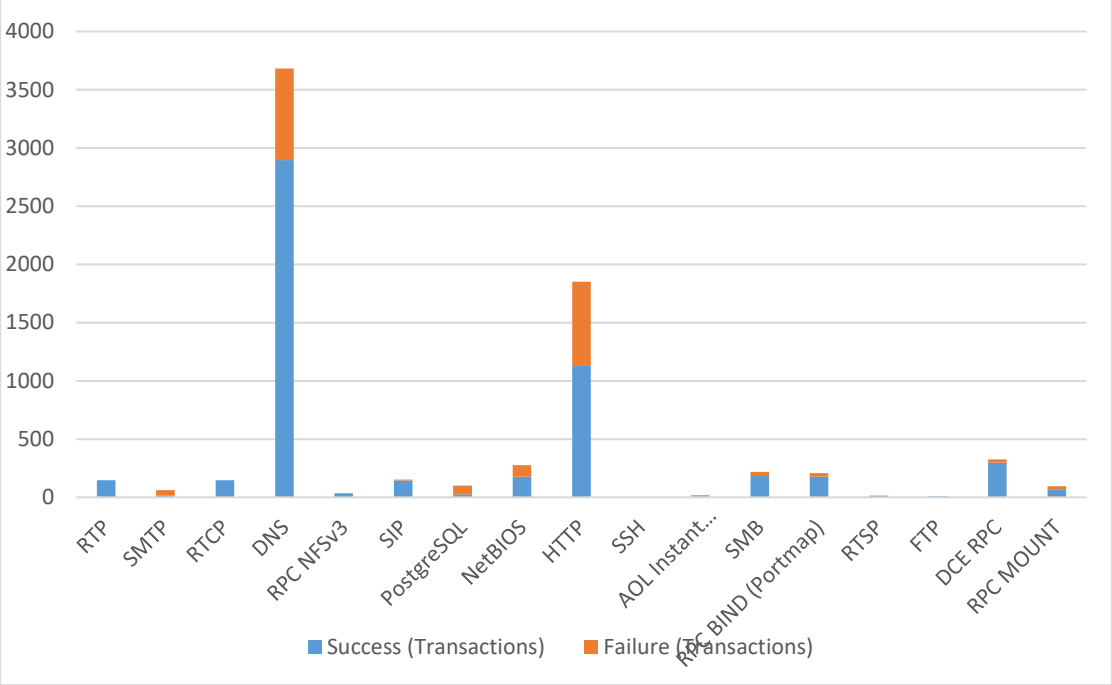
Appendix 4. Sophos UTM first test detailed detailed strike category assessment



Appendix 5. Sophos UTM second test detailed strike category assessment

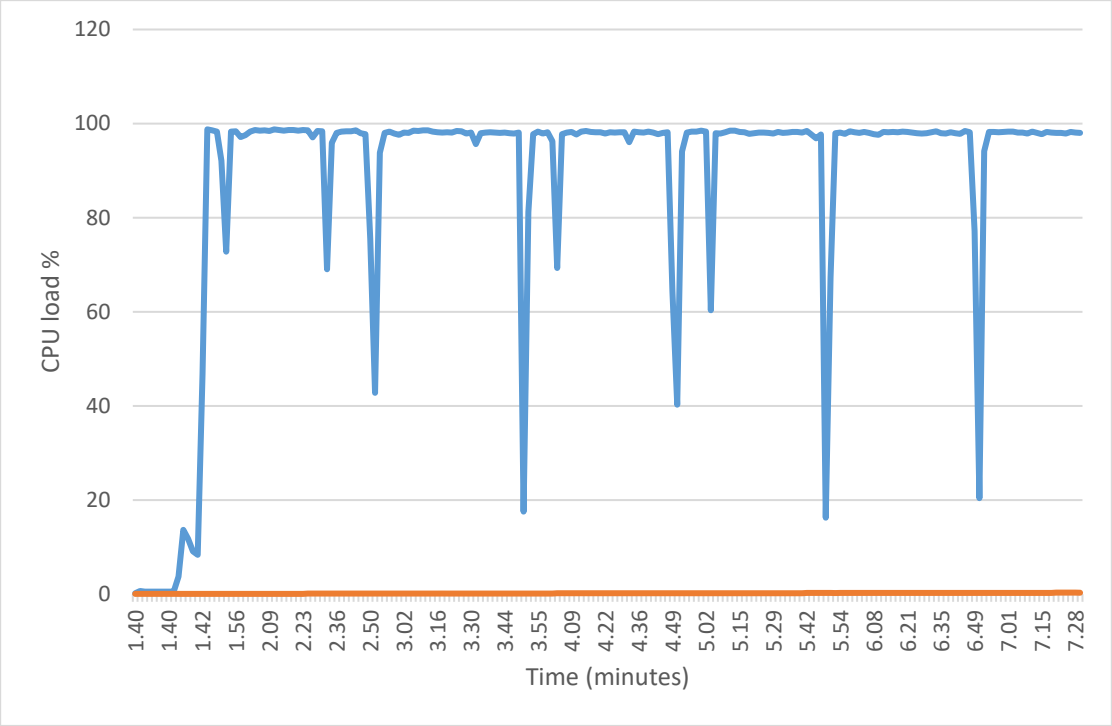


Appendix 6. Sophos UTM second test application transactions summary

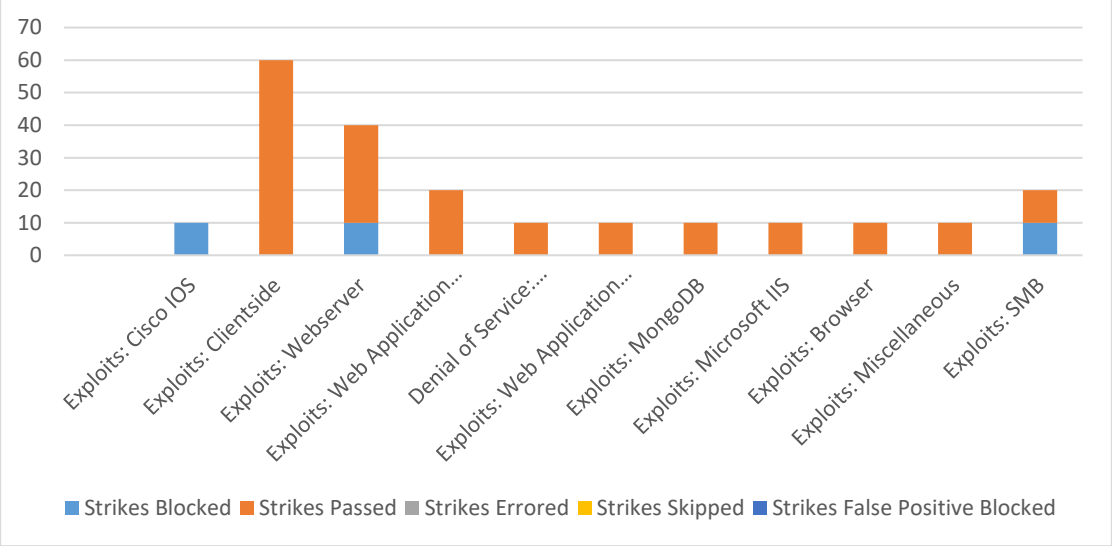


Protocol	Success (Transactions)	Failure (Transactions)
RTP	147	0
SMTP	16	46
RTCP	147	0
DNS	2896	787
RPC NFSv3	35	1
SIP	145	8
PostgreSQL	27	73
NetBIOS	177	98
HTTP	1128	723
SSH	0	1
AOL Instant Messenger	10	8
SMB	184	34
RPC BIND (Portmap)	176	31
RTSP	13	3
FTP	9	0
DCE RPC	299	27
RPC MOUNT	65	30

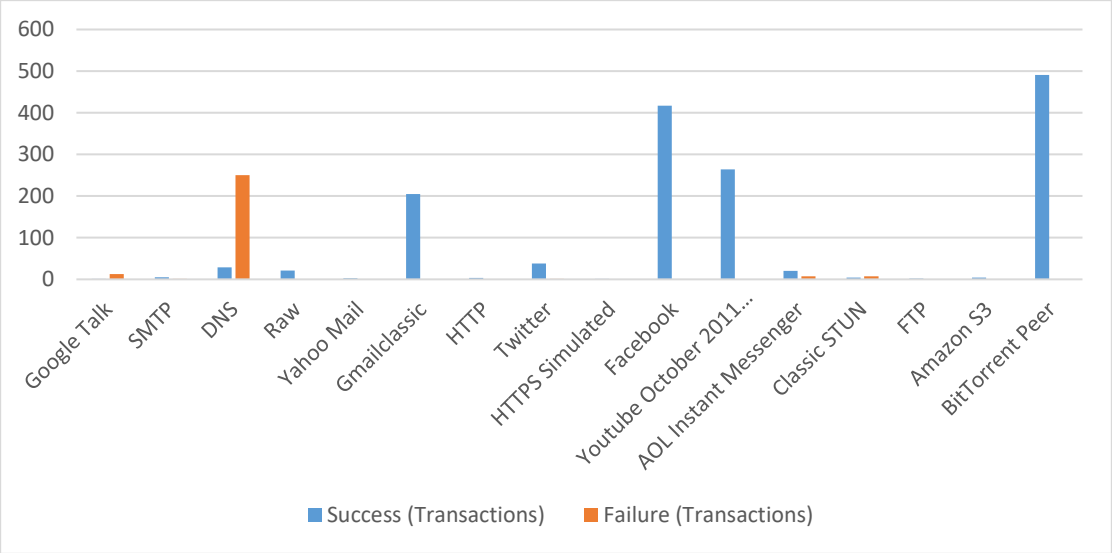
Appendix 7. Sophos UTM second test snort CPU load



Appendix 8. Sophos UTM third test detailed detailed strike category assessment

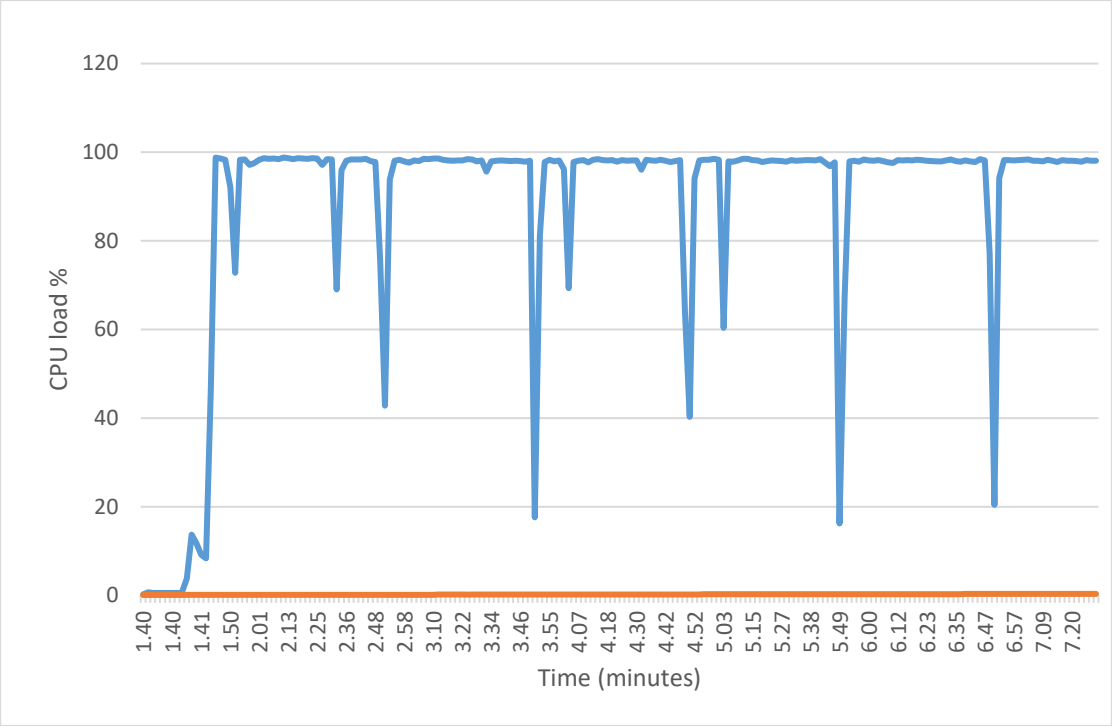


Appendix 9. Sophos UTM third test application transactions summary

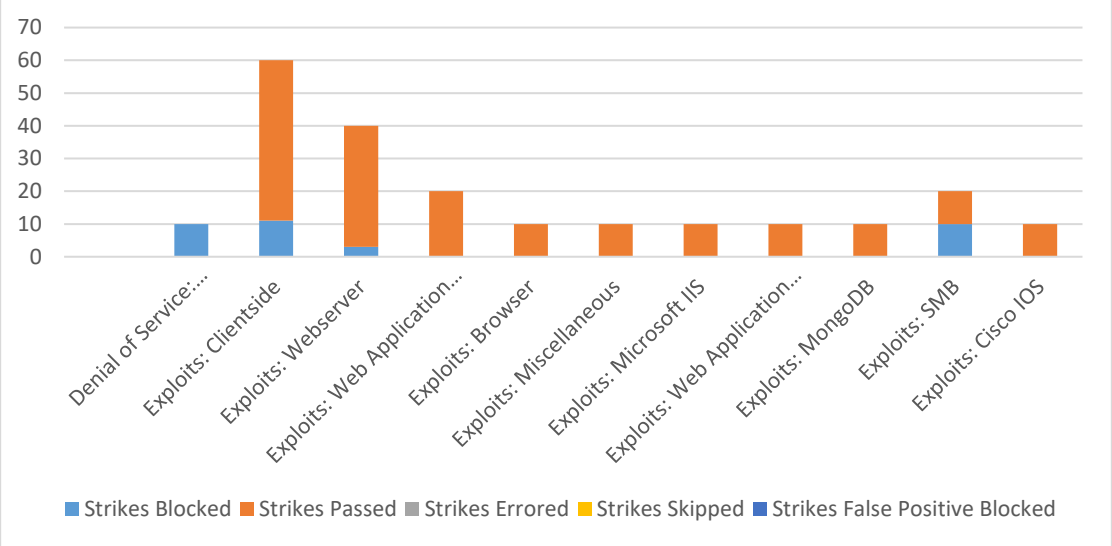


Protocol	Success (Transactions)	Failure (Transactions)
Google Talk	145	13
SMTP	5163	1
DNS	29 126	250
Raw	21	0
Yahoo Mail	2856	0
Gmailclassic	205	0
HTTP	3493	0
Twitter	38 094	1
HTTPS Simulated	1499	0
Facebook	417	0
Youtube October 2011 (Deprecated)	264	0
AOL Instant Messenger	20 119	7
Classic STUN	4489	7
FTP	1637	0
Amazon S3	4	0
BitTorrent Peer	491	0

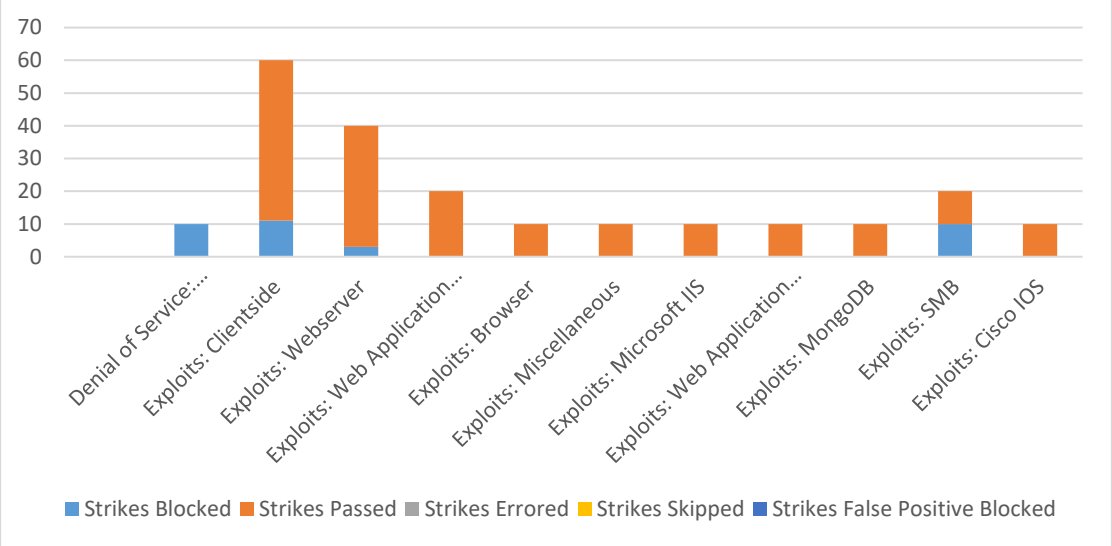
Appendix 10. Sophos UTM third test snort CPU load



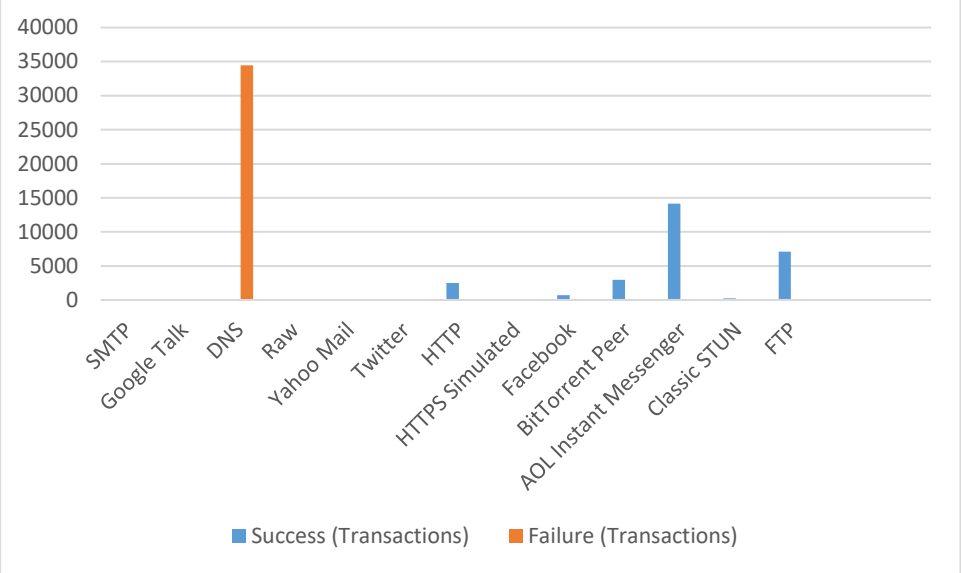
Appendix 11. Cisco Asa Firepower first test detailed strike category assessment



Appendix 12. Cisco Asa Firepower second test detailed strike category assessment

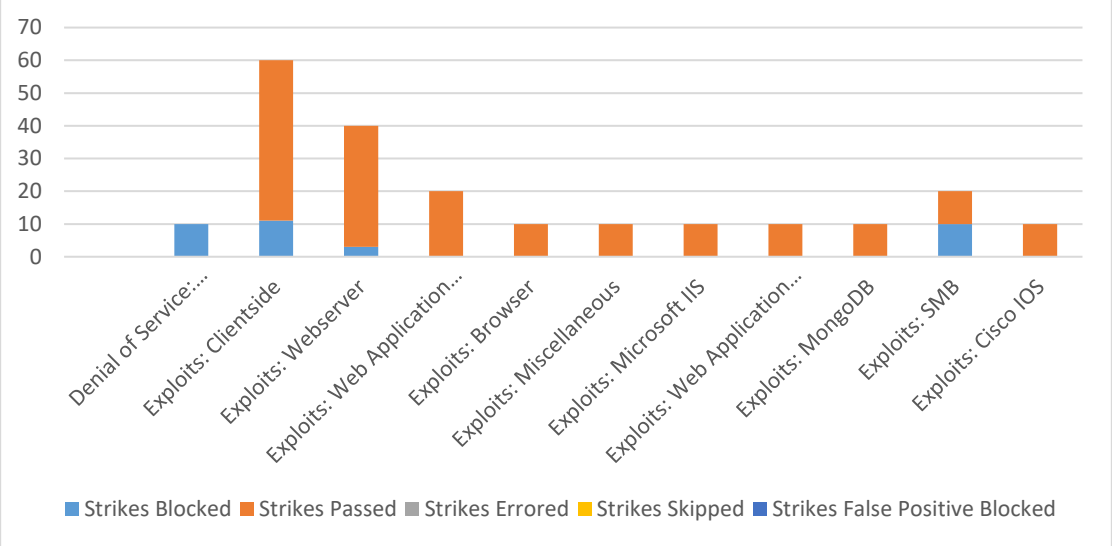


Appendix 13. Cisco Asa Firepower second test application transactions summary

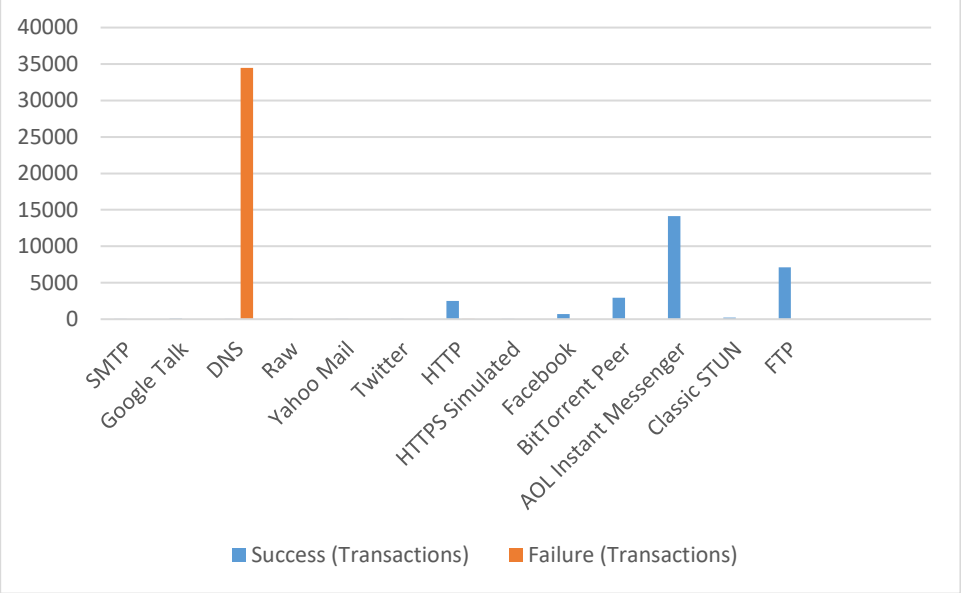


Protocol	Success (Transactions)	Failure (Transactions)
RTP	600	0
SMTP	7	0
RTCP	600	0
DNS	31	40 415
RPC NFSv3	591	0
PostgreSQL	2	0
NetBIOS	10 421	0
HTTP	9	13
SSH	118	0
AOL Instant Messenger	1291	0
SMB	1105	10 273
RPC BIND (Portmap)	1182	0
RTSP	3	0
DCE RPC	0	238
RPC MOUNT	591	0

Appendix 14. Cisco Asa Firepower third test detailed strike category assessment

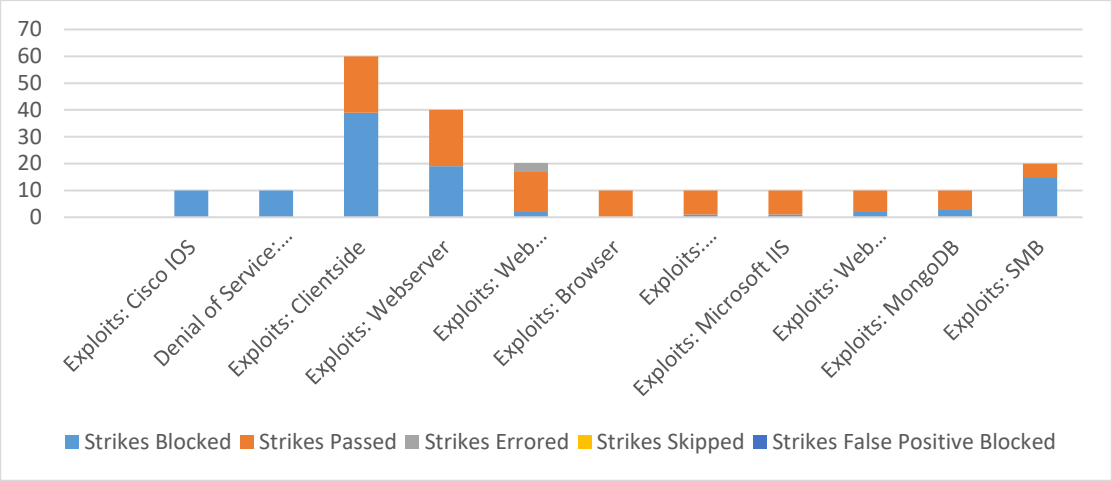


Appendix 15. Cisco Asa Firepower third test application transactions summary

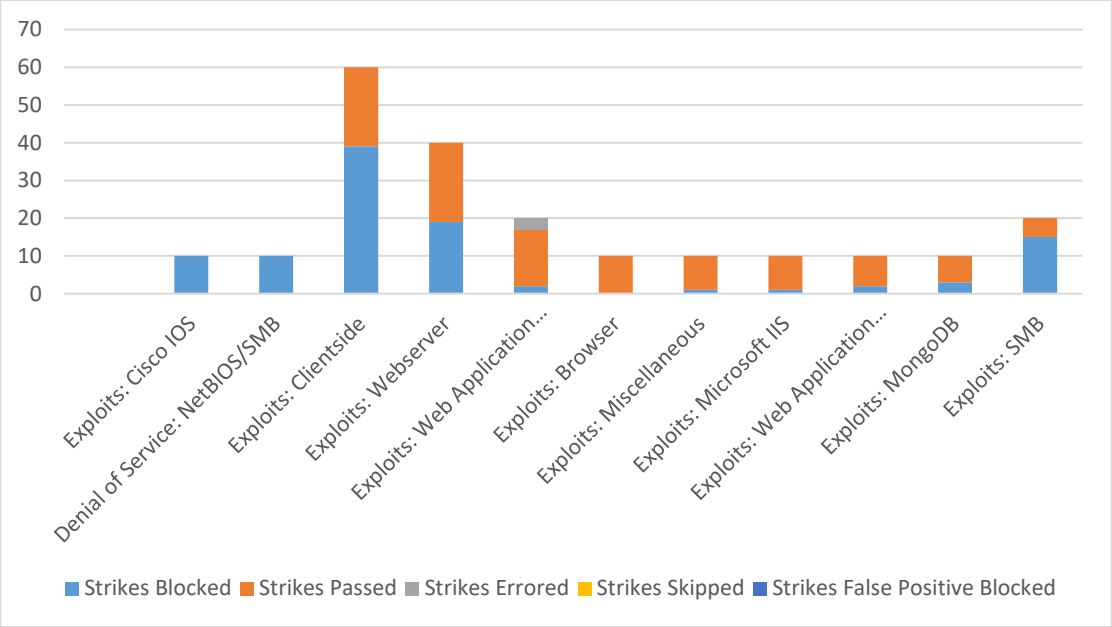


Protocol	Success (Transactions)	Failure (Transactions)
SMTP	6	0
Google Talk	77	0
DNS	57	34 461
Raw	9	0
Yahoo Mail	12	0
Twitter	44	0
HTTP	2488	7
HTTPS Simulated	16	0
Facebook	715	0
BitTorrent Peer	2954	0
AOL Instant Messenger	14 157	0
Classic STUN	238	0
FTP	7115	0

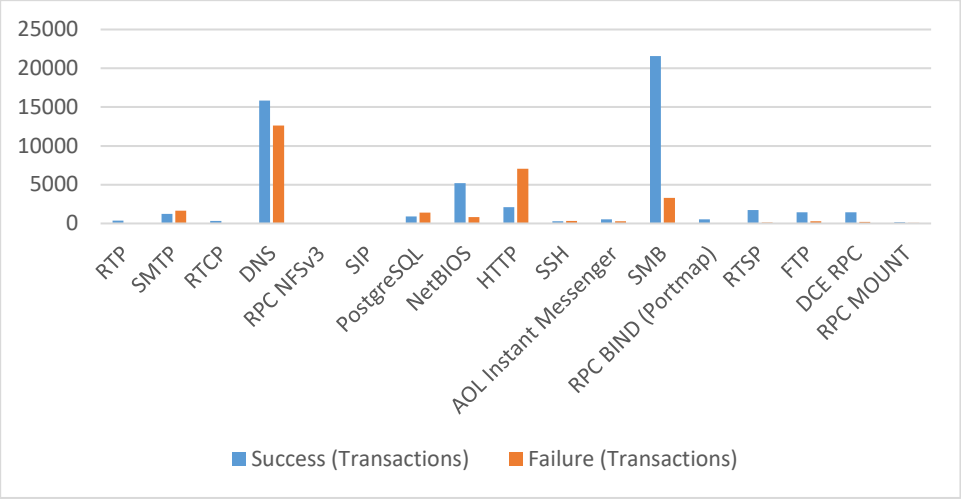
Appendix 16. OPNsense first test detailed strike category assessment



Appendix 17. OPNsense second test detailed strike category assessment

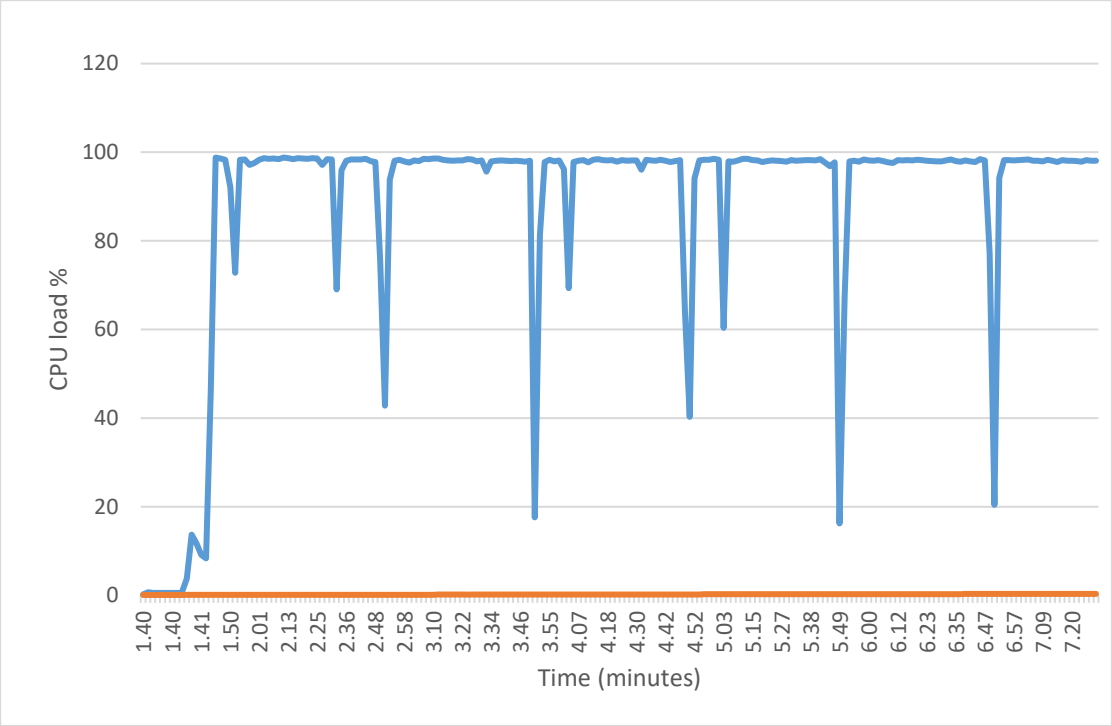


Appendix 18. OPNsense second test application transactions summary

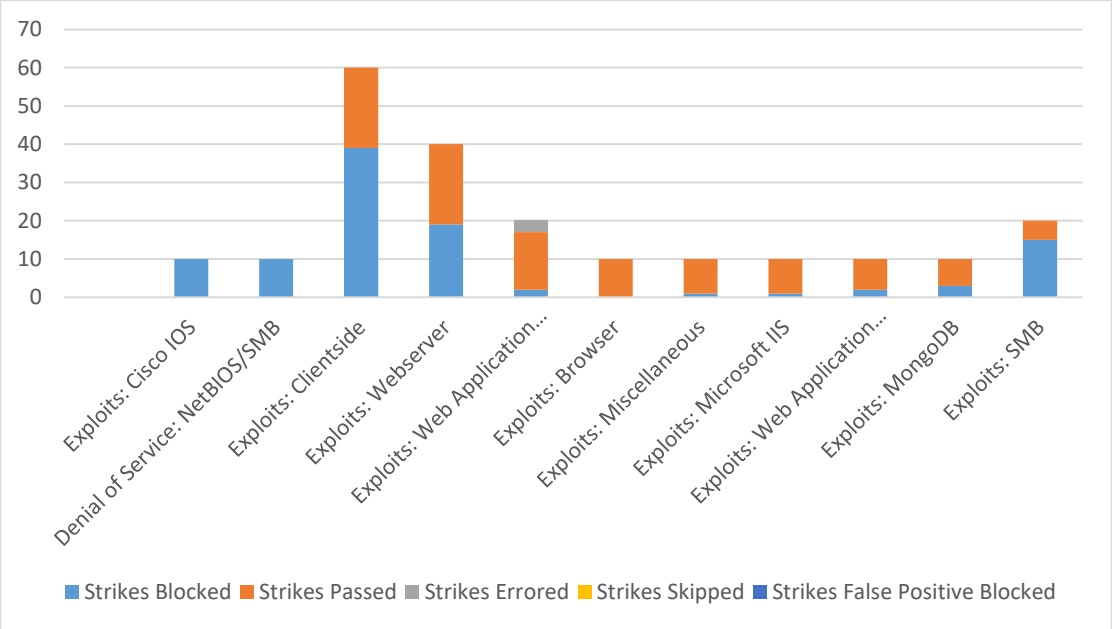


Protocol	Success (Transactions)	Failure (Transactions)
RTP	348	0
SMTP	1226	1627
RTCP	339	0
DNS	15 819	12 603
RPC NFSv3	32	22
SIP	38	29
PostgreSQL	911	1381
NetBIOS	5188	824
HTTP	2119	7047
SSH	289	340
AOL Instant Messenger	538	297
SMB	21 575	3285
RPC BIND (Portmap)	526	2
RTSP	1734	124
FTP	1452	267
DCE RPC	1438	208
RPC MOUNT	144	98

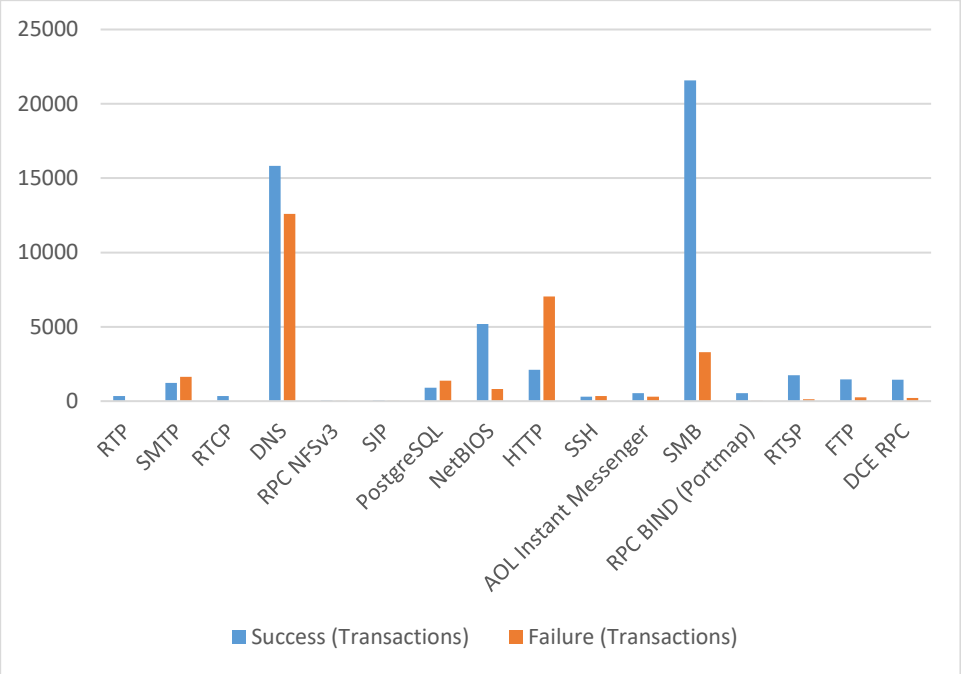
Appendix 19. OPNsense second test suricata CPU load



Appendix 20. OPNsense third test detailed strike category assessment



Appendix 21. OPNsense third test application transactions summary



Protocol	Success (Transactions)	Failure (Transactions)
Google Talk	2275	2068
SMTP	2447	1529
DNS	21 389	12 285
Raw	0	4
Yahoo Mail	5058	1819
Gmailclassic	4	123
HTTP	1164	1212
Twitter	678	81
HTTPS Simulated	2771	1214
Facebook	165	499
Youtube October 2011 (Deprecated)	169	169
AOL Instant Messenger	9874	3774
Classic STUN	4122	34
FTP	3764	632
Amazon S3	110	260
BitTorrent Peer	346	1011

Appendix 22. OPNSense third test suricata CPU load