

Black Book

ixia

Edition 10

Network Security

NETWORK SECURITY

Your feedback is welcome

Our goal in the preparation of this Black Book was to create high-value, high-quality content. Your feedback is an important ingredient that will help guide our future books.

If you have any comments regarding how we could improve the quality of this book, or suggestions for topics to be included in future Black Books, contact us at ProductMgmtBooklets@ixiacom.com.

Your feedback is greatly appreciated!

Copyright © 2014 Ixia. All rights reserved.

This publication may not be copied, in whole or in part, without Ixia's consent.

RESTRICTED RIGHTS LEGEND: Use, duplication, or disclosure by the U.S. Government is subject to the restrictions set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 and FAR 52.227-19.

Ixia, the Ixia logo, and all Ixia brand names and product names in this document are either trademarks or registered trademarks of Ixia in the United States and/or other countries. All other trademarks belong to their respective owners. The information herein is furnished for informational use only, is subject to change by Ixia without notice, and should not be construed as a commitment by Ixia. Ixia assumes no responsibility or liability for any errors or inaccuracies contained in this publication.

Contents

How to Read this Book	vii
Dear Reader.....	viii
Network Security.....	1
Test Methodologies for Known Vulnerabilities	25
Test Case: Measuring the Security Effectiveness of Intrusion Prevention Systems.....	31
Test Methodologies for DoS and DDoS	41
Test Case: Application Forwarding Performance under DoS Attacks.....	57
Test Case: Mitigation of TCP SYN DDoS attack.....	69
Test Case: Mitigation of ICMP Fragments DDoS Flooding attack.....	81
Test Case: Mitigation of IP Short Fragments Flooding DDoS attack.....	85
Test Case: Mitigation of a DDoS MIX Pattern Using Even Test Objective Distribution Over Same Test Interface	89
Test Case: Mitigation of a DDoS MIX Pattern Using Uneven Test Objective Distribution over Same Test Interface.....	95
Test Methodologies: IPsec VPN.....	101
Test Case: IPsec - Data Forwarding Performance	115
Test Case: IPsec - Tunnel Capacity Test.....	137
Test Case: IPsec Quick Test - RFC 2544 Throughput	155
Test Case: IPsec Quick Test – Tunnel Setup Rate	171
Test Case: IPsec Quick Test – Tunnel Capacity	187
Appendix A: Configuring IP and Network Settings.....	203
Appendix B: Configuring TCP Parameters.....	205
Appendix C: Configuring HTTP Servers.....	207
Appendix D: Configuring HTTP Clients	209
Appendix E: Setting the Test Load Profile and Objective	211
Appendix F: Adding Test Ports and Running Tests	213
Appendix G: StrongSwan IPsec VPN Gateway Sample Configuration.....	215
Appendix H: Application Forwarding Performance under DoS Attacks with Network Impairment Added	217
Contact Ixia	225

How to Read this Book

The book is structured as several standalone sections that discuss test methodologies by type. Every section starts by introducing the reader to relevant information from a technology and testing perspective.

Each test case has the following organization structure:

Overview	Provides background information specific to the test case.
Objective	Describes the goal of the test.
Setup	An illustration of the test configuration highlighting the test ports, simulated elements and other details.
Step-by-Step Instructions	Detailed configuration procedures using Ixia test equipment and applications.
Test Variables	A summary of the key test parameters that affect the test's performance and scale. These can be modified to construct other tests.
Results Analysis	Provides the background useful for test result analysis, explaining the metrics and providing examples of expected results.
Troubleshooting and Diagnostics	Provides guidance on how to troubleshoot common issues.
Conclusions	Summarizes the result of the test.

Typographic Conventions

In this document, the following conventions are used to indicate items that are selected or typed by you:

- **Bold** items are those that you select or click on. It is also used to indicate text found on the current GUI screen.
- *Italicized* items are those that you type.

NETWORK SECURITY

Dear Reader

Ixia's Black Books include a number of IP and wireless test methodologies that will help you become familiar with new technologies and the key testing issues associated with them.

The Black Books can be considered primers on technology and testing. They include test methodologies that can be used to verify device and system functionality and performance. The methodologies are universally applicable to any test equipment. Step by step instructions using Ixia's test platform and applications are used to demonstrate the test methodology.

This tenth edition of the black books includes twenty two volumes covering some key technologies and test methodologies:

Volume 1 – Higher Speed Ethernet

Volume 12 – IPv6 Transition Technologies

Volume 2 – QoS Validation

Volume 13 – Video over IP

Volume 3 – Advanced MPLS

Volume 14 – Network Security

Volume 4 – LTE Evolved Packet Core

Volume 15 – MPLS-TP

Volume 5 – Application Delivery

Volume 16 – Ultra Low Latency (ULL) Testing

Volume 6 – Voice over IP

Volume 17 – Impairments

Volume 7 – Converged Data Center

Volume 18 – LTE Access

Volume 8 – Test Automation

Volume 19 – 802.11ac Wi-Fi Benchmarking

Volume 9 – Converged Network Adapters

Volume 20 – SDN/OpenFlow

Volume 10 – Carrier Ethernet

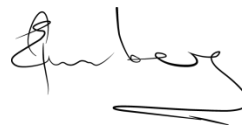
Volume 21 – Network Convergence Testing

Volume 11 – Ethernet Synchronization

Volume 22 – Testing Contact Centers

A soft copy of each of the chapters of the books and the associated test configurations are available on Ixia's Black Book website at <http://www.ixiacom.com/blackbook>. Registration is required to access this section of the Web site.

At Ixia, we know that the networking industry is constantly moving; we aim to be your technology partner through these ebbs and flows. We hope this Black Book series provides valuable insight into the evolution of our industry as it applies to test and measurement. Keep testing hard.



Errol Ginsberg, Acting CEO

Network Security

Test Methodologies

Network security is essential for homes, government organizations, and enterprises of all sizes. The number and types of attacks are enormous and the devices used to defend against them are necessarily complex. This book provides an overview of network security and covers test methodologies that can be used to assess the effectiveness and performance impact of IPS/IDS, UTMs, and new generation firewalls while they are attacked using threats that include DoS/DDoS, exploits based on known vulnerabilities, and malware. The last section of the book is dedicated on IPsec VPN test methodologies.

Network Security

Security is a discipline concerned with protecting networks and computer systems against threats such as exploits, malware, data leakage, spam, and DoS attacks, as well as ensuring trusted access through mechanisms like IPsec or SSL. To defend against threats, and to prevent unintended data leakage, enterprises have deployed security devices of all types.

Network security devices include one or more security functions, including firewall, intrusion prevention/detection systems (IPS/IDS), data leakage prevention (DLP), and content security filtering functions (anti-spam, antivirus, URL filtering). Those functions start to be integrated more often in what is called *Unified Threat Management* system or *New Generation Firewall*. Each type of device, and the unified threat management (UTM) systems that combine them into one system, requires continuous testing to ensure that the devices are effective, accurate, and productive.

Securing the networks is essential for homes, government organizations, and enterprises of all sizes. The number and types of attacks continues to grow at an alarming rate and the devices used to defend against them are necessarily complex. The complexity and effectiveness of attacks continues to increase, positioning the network security as a growing concern for end-users and enterprises of all sizes.

This book provides an overview of network security and covers test methodologies that can be used to assess the effectiveness, accuracy, and performance of such devices while they are inspecting legitimate traffic and malicious traffic. Lastly, a set of IPsec VPN test methodologies covers test cases that can be used to address the performance of IPsec control plane and data plane protocols, as well as to measure the security effectiveness of UTM and new generation firewalls that combine IPS/IDS and VPN Gateways on same device.

The Current State of Network Security

There has been an explosion of security threats in recent years. According to the 2009 Annual Report from Panda Labs:

"In 2009, over 25 million different unique malware programs were identified, more than all the malware programs ever created in all previous years combined."

The breakdown of the types of malware programs found by Panda Labs is shown in Figure 1. These categories are explained in this book.

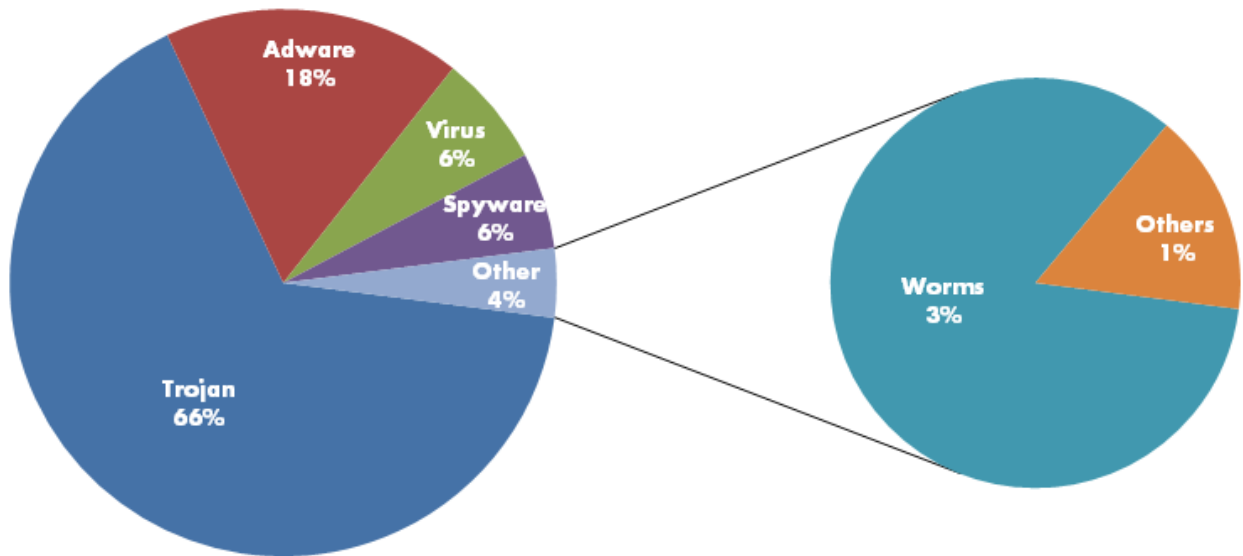


Figure 1. Breakdown of malware types, 2009

Hacking has mutated from a hobby to a successful business. 78% of malware attacks export user data, and 70% of the targets were banks. It is estimated that companies lose between 0.5 and 2.5% of their revenues because of security-related losses and downtime.

In 2008, McAfee Security surveyed 800 CIOs worldwide about security losses. The estimated loss from the surveyed companies was USD 4.6 billion; USD 600 million was spent repairing the damage from data/network breaches. The biggest threats were from employees who had been laid off and attacks from outside the company. McAfee estimated that total worldwide costs in 2008 would top USD 1 trillion.

The 2009 Ponemon Institute Annual Study, as reported in Network World's January 25, 2010 issue, found that the average data breach cost USD 204 per compromised customer record. The total cost to a company for each breach averaged USD 6.75 million.

During the last few years, the cumulative number of vulnerabilities has increased dramatically, as shown in Figure 2.

NETWORK SECURITY

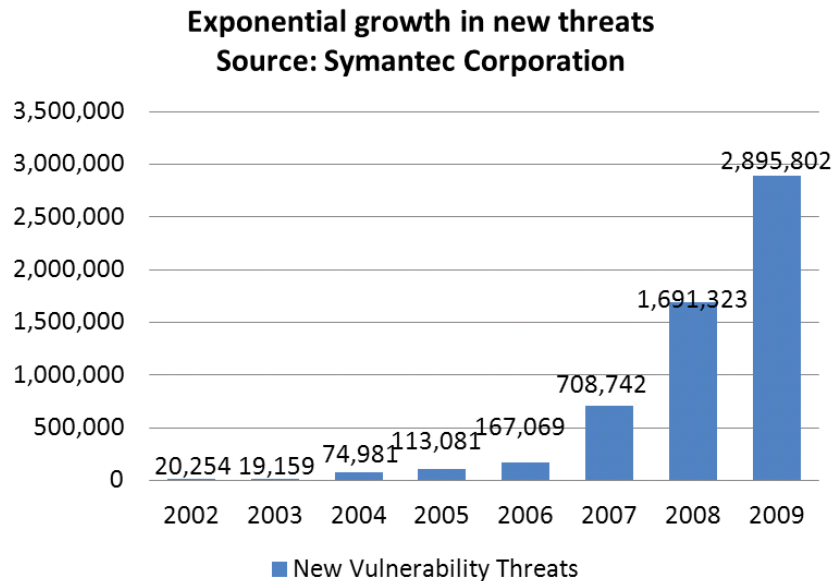


Figure 2. Growth in new threats

The number of vulnerabilities discovered in applications is far greater than the number discovered in operating systems. As a result, more exploitation attempts are recorded on application programs. The most popular applications for exploitation tend to change over time because the rationale for targeting a particular application often depends on factors like prevalence or the inability to effectively patch. Browsers and client-side applications that can be invoked by browsers seem to be consistently targeted, taking advantage to the current trend wherein trusted Web sites are converted into malicious servers.

Worldwide, there has been a significant increase over the past three years in the number of people discovering zero-day vulnerabilities, as measured by multiple independent teams discovering the same vulnerabilities at different times. Zero-day vulnerabilities are those not found until a service is deployed.

The Source of the Problem

But, who is to blame for the vulnerabilities that malware takes advantage of? The Internet is something that we all want—the ability to publish and find information, the ability to buy and sell products, the ability to communicate with others. The vast interconnection made possible by the Internet provides the avenue for malicious action.

The major avenue of attack is through flawed software. That is, software that is outright broken or sloppily written. A typical example is the buffer overflow flaw, in which a programmer invites a user's response, but does not compare the length of the response against the amount of storage set aside for the response. A carefully constructed, overly long, malicious response can be used crash the software, or cause it to execute arbitrary computer code—code designed to steal data or embed other attacks.

NETWORK SECURITY

Flaws are classified as either known or unknown, zero-day vulnerabilities. Known vulnerabilities are published, allowing authors to issue fixes and security vendors to update software. Zero-day vulnerabilities are potentially more harmful, associated with newly published programs or offered Web services. Such vulnerabilities may be visible for days or weeks until patched.

Network, server, and client misconfiguration offers another avenue for hacking. Network elements, such as routers and home gateways, come with a default administrator password, passwords that often never change. A hacker with access to a router can cause all traffic through the router to be sent through its own server, allowing 'person-in-the-middle' attacks.

Similarly, misconfigured servers can allow hackers to disable or modify Web sites, inserting code of its own choosing. Such code is usually intended to steal data from associated databases.

Finally, many of us are to blame. We are often not careful enough, gullible or too trusting—allowing attackers to get us to cooperate with their plans.

The Damage

The damage from successful network security attacks can take many forms:

Loss of data. This consists not only of just financial data, such as credit card numbers, but also includes customer lists, intellectual property, and product development and marketing plans.

Loss of time. It can take a great deal of time to recover from a security attack, or even from the suspicion of an attack. Data may need to be recovered or reconstructed and systems extensively checked.

Monetary loss. This is often preceded by the theft of data.

Disabled or crippled services. Protesters and some governments may seek to disable offending Web sites. Hackers may be purely malicious in their intent.

Legal exposure. Any of the previous items may expose an enterprise to law suits for loss of data or money entrusted to them.

Classification of Security Attacks

User-Involved Attack Mechanisms

Computer users are the primary avenue used in security attacks. The most frequent methods include the following:

- **E-mail.** In addition to spam, e-mails can contain attachments that are malicious executable programs or links to infected Web sites. Waves of targeted e-mail attacks, often called spear-phishing, are exploiting client-side vulnerabilities in commonly used programs such as Adobe® PDF Reader®, QuickTime, Adobe® Flash® and Microsoft® Office. This is currently the primary initial infection vector used to compromise computers that have Internet access.
- **Web.** Those same client-side vulnerabilities are exploited by attackers when users visit infected Web sites. Because the visitors feel safe downloading documents from the trusted sites, they are easily fooled into opening documents, music, and video that exploit client-side vulnerabilities. Some exploits do not even require the user to open documents. Simply accessing an infected Web site is all that is needed to compromise the client software. Web sites can be dangerous in several ways:

Masquerading as valid Web sites collecting financial and personal information.

Infected through content injected from associated Web sites. The average commercial Web page contains content from more than 100 sources—advertising, tracking, and content. One or more of those sources may have been compromised and may insert code that is used to collect and send data to a third party.

Present false information. For example, a Web page advertisement might suggest that a user's computer is infected with a virus, inviting the user to click on a virus scanning program, which actually infects the computer.

- **FTP.** FTP is frequently used to download executable programs. These programs may be innocently or maliciously infected.
- **Instant Messaging (IM).** Instant messaging programs now provide mechanisms for passing executable programs and Web links, providing a means of infecting computers and revealing information.
- **Peer-to-peer (P2P).** P2P environments are often used to share software, which may be similarly infected.
- **Gaming.** Social interaction with other players may invite e-mail or IM communications. Games themselves, when executed on a user's computer, may be the source of infections. Games that must be run in administrator mode or use ActiveX or JavaScript are especially suspected.
- **Software updates.** Software vendors are increasingly updating their software over the Internet, using Web pages, or dedicated, resident programs. Malicious parties may substitute their own software, or infect the updates before they are downloaded.
- **People.** End-users are frequently at fault for the following reasons:

NETWORK SECURITY

- **Using poor passwords.** Using easy to guess passwords or reusing the same set of passwords over and over again.
- **Inconsistently updating their software.** Many attacks take advantage of known operating system and application vulnerabilities. Software vendors usually offer software updates that plug these vulnerabilities, but they must be applied by the end-user.
- **Getting too personal.** Online groups often ask for personal information, for example, spouse, children, and pet names. This information may be used for identity theft or password guessing.
- **Being too trusting.** Friends and other acquaintances may send us software or Web sites, and we frequently trust them because we know them. They may have been duped or the message may have been falsified.
- **Inconsistent application of security software.** Computer security can be confusing for a computer user, including personal and corporate firewalls, anti-virus software, anti-spam software, and browser and e-mail protection. All types of protection must be applied.
- **Engaging in wishful thinking:** "It will not happen to me."

Web vulnerabilities comprise 49% of the total number of those reported. The cumulative number of reported Web vulnerabilities is more than 20,000. Attacks against Web applications constitute more than 60% of the total attack attempts observed on the Internet. These vulnerabilities are being exploited widely to convert trusted Web sites into malicious Web sites serving content that contains client-side exploits.

Three types of vulnerabilities predominate:

- **Cross-site scripting (XSS).** This type of exploit inserts HTML or other Web content into Web pages before they are displayed to the user. Such code is often used to steal personal information or to direct the viewer to a different Web site.
- **SQL injection.** This type of exploit extracts information from a database. For example, users might be prompted for their account information; the Web application may be expecting a simple answer such as *John Smith* and use that name in an SQL query of the form:

statement = "SELECT * FROM users WHERE name = " + userName + " ;"

However, a hacker answering by typing:

' or '1'='1

will generate the following query *statement*:

SELECT * FROM users WHERE name = " OR '1'='1' ;

The result would be that the statement would return information for all users in the database. Proper care in programming would prevent SQL injection attacks, but many Web sites are still vulnerable.

- **File includes.** This vulnerability is similar to SQL injection in that it takes advantage of unchecked user input. Such input may be used with Web sites that use PHP or Java. The unchecked user input is used to include additional code from a hacker's site using file include facilities in the Web language.

Figure 3 shows the most common malicious software in common Web downloads.

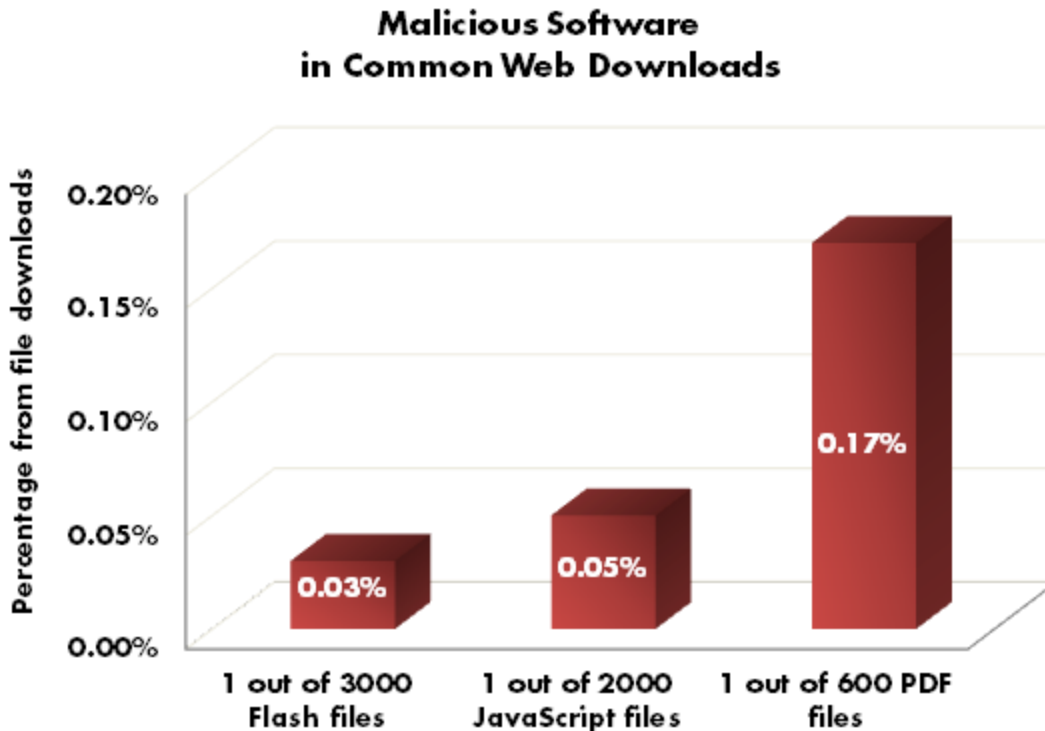


Figure 3. Malicious software in common Web downloads

Web application vulnerabilities, such as SQL injection and cross-site scripting flaws account for more than 80% of the vulnerabilities reported. Despite the enormous number of attacks and despite widespread publicity about these vulnerabilities, most Web site owners fail to scan effectively for the common flaws. They become unwitting tools used by criminals to infect the visitors that trusted those sites to provide a safe Web experience.

Network-Level Attack Mechanisms

A number of attacks are mounted without user involvement. The Internet depends on a number of services accessible to everyone: Web, DNS, FTP, SMTP, POP, IMAP, and SIP to name just a few. The server software used for these services, plus the many plug-ins that are used in conjunction with the services, are an attractive target for hackers. All software has vulnerabilities and hackers are able to find them and exploit them for theft or other nefarious purposes.

Sites that offer their users remote access may rely solely on user passwords. Automated 'robots' may try long lists of possible passwords in order to gain access. They may use information that the user has provided to others, for example, see the *Getting Too Personal* earlier.

NETWORK SECURITY

Denial of service attacks are another network-level threat in which the attacker uses large numbers of hijacked computers to send malicious traffic to a Web or other server. The purpose of the attack is to disable the service partially or completely.

Sources of Vulnerabilities

Vulnerabilities are a result of software flaws, flaws that fail to anticipate all possible conditions, especially unusual user input. Software flaws exist in all software; most are innocuous, but many provide the means for security penetrations. Figure 4 shows the distribution of known vulnerabilities across the top 10 software vendors, as reported in the *X-Force 2009 Trends and Risk Report*.

Ranking	Vendor	Disclosures
1.	Apple	3.8%
2.	Sun	3.3%
3.	Microsoft	3.2%
4.	IBM	2.7%
5.	Oracle	2.2%
6.	Mozilla	2.0%
7.	Linux	1.7%
8.	Cisco	1.5%
9.	Adobe	1.4%
10.	HP	1.2%

Table 3: Vendors with the Most Vulnerability Disclosures, 2009

Figure 4. Top 10 vendors with most vulnerabilities disclosed in 2009

Overall, more than 50% of the vulnerabilities reported in 2008 remained unpatched in 2009. Among the top 10 list in Figure 4, the record is somewhat better, with only 21% remaining unpatched.

Although the total number of vulnerabilities continues to grow, the rate of growth has stabilized over the last several years. Figure 5 shows the rate of disclosed vulnerabilities from 2000 to 2009, according to X-Force.

NETWORK SECURITY

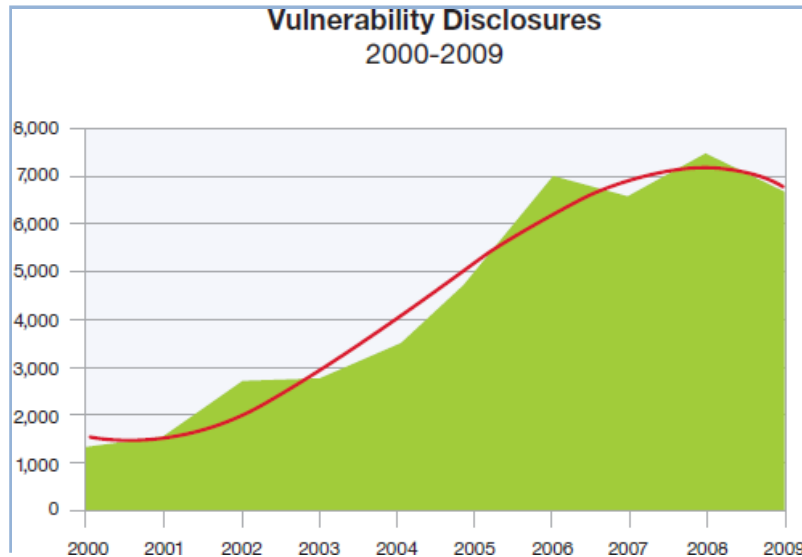


Figure 5. Vulnerabilities disclosed from 2000 to 2009

The number of vulnerabilities discovered in each of the years since 2006 has been between 6,000 and 7,000. The number of vulnerabilities may be stabilizing because of aggressive patching by software vendors. For example, the Conficker botnet managed to assemble more than 6,500,000 computers into the world's largest computing network. The botnet was used only once in April 2009. As of 2010, it is no longer considered as a major threat because of the patching of the underlying vulnerabilities that enabled the growth of the botnet.

Nevertheless, the problem of discovering and verifying new vulnerabilities is a very large industry problem. Some security companies receive more than 55,000 new samples **per day**.

Malware

Malware is the term used to describe the entire gamut of malicious software. For the purpose of this discussion, we will break them down into six categories:

- Viruses
- Worms
- Trojans
- Rootkits
- Spyware
- Malicious adware/scareware

Although we can distinguish these types, modern malware is very often hard to categorize—blending multiple types of attacks.

Viruses

The term virus is often used instead of malware, but actually refers to a computer program that infects a computer and is spread by user action. A virus typically attaches itself to another program. User actions include e-mail and physical distribution through CD, DVD, or USB drive.

NETWORK SECURITY

Viruses often establish themselves on shared network file systems that may be accessed from multiple computers.

Worms

Worms are self-spreading programs that take advantage of security vulnerabilities. They spread themselves to other network nodes without any user interaction. Worms typically stand alone and do not need to attach themselves to other programs. They may consume bandwidth, or corrupt or modify files. The first significant worm was the Morris worm, which was released on November 2, 1988. It took advantage of known vulnerabilities in Unix sendmail, finger, and rsh/rexec, as well as weak passwords. It was originally intended to measure the size of the Internet, but a coding error caused it to infect computers multiple times, rendering them too busy to be useful.

More recently, the Koobface worm has spread to more than 3 million computers. It is spread through social networking site invitations and friend e-mails.

Trojans

Trojans are programs that appear harmless, but hide malicious functions. These functions are often remotely controlled by central computers. They are particularly insidious because they may do nothing for long periods of time, or only intermittently. They may do the following:

- Make the computer available as part of a network of remotely controlled computers, called a *botnet*.
- Steal personal user information, either by scanning local information or by injecting code into Web forms and e-mail.
- Install other malware.
- Download or upload files, wasting computer storage space and network bandwidth.
- Modify or delete files.
- Log keystrokes to discover passwords and other information.
- Transmit the contents of a user's screen.

Banking trojans, used to steal credit card and banking information, increased 200% in 2009. The Zeus trojan is an example in point, collecting financial information through code injection in specific Web pages. What makes this trojan particularly nasty is that toolkits are readily available for only USD 700 that construct a customized trojan, as shown in Figure 6.

NETWORK SECURITY

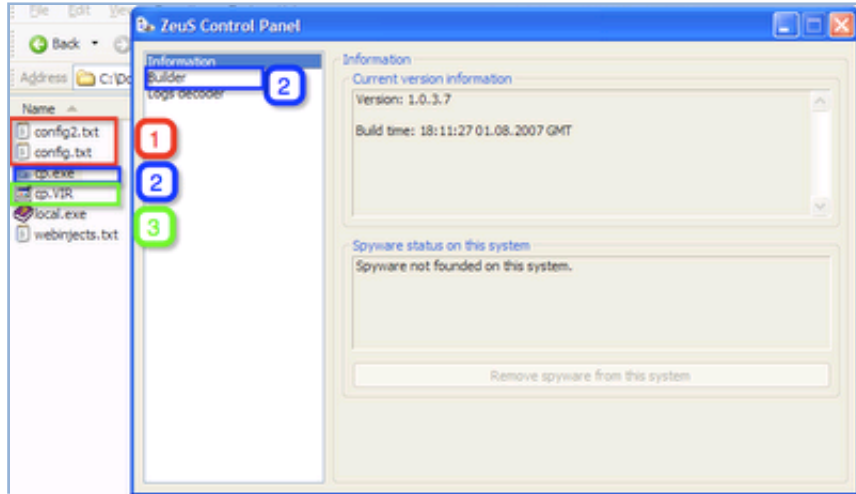


Figure 6. Zeus trojan toolkit

The toolkit generates a unique version each time it is used, making it difficult for anti-virus vendors to detect the trojan. Figure 7 tracks the number of unique Zeus trojan derivatives over time, according to the *2009 Symantec Security Report*.

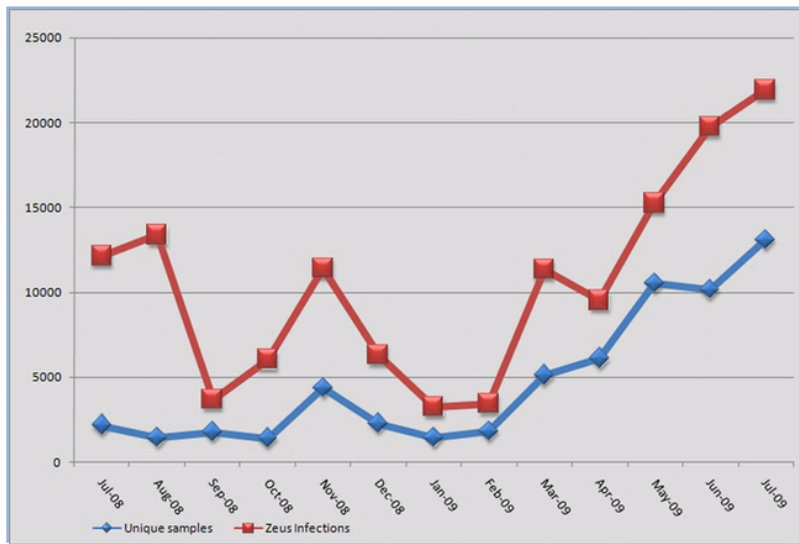


Figure 7. Zeus trojan infections

Rootkits

Rootkits are self-obscuring programs, replacing normal operating system files. In doing so, they disable security software packages that might discover them. As part of the operating system, they can perform any number of functions. For example, in Unix-based systems, they can replace the *login* program, capturing valuable user passwords for later use. Rootkits exist for a wide variety of operating systems, including Microsoft Windows, Linux, Mac OS, and Solaris.

Spyware

Spyware is a type of hidden malware that collects and forwards user and computer information. It can be used to collect various types of personal information, such as Internet surfing habits.

NETWORK SECURITY

Spyware can also interfere with user control of their computer in other ways, such as installing additional software and redirecting Web browser activity. Spyware has been known to change computer settings, resulting in slow connection speeds, altered home pages and loss of Internet connectivity, or loss of functionality of other programs.

Spam

Spam includes any type of unwanted message. Spam is usually delivered by e-mail and in most cases, seeks to sell something through an included link. Figure 8 is a list of the most popular spam subject lines.

Subject Line	%
You've received an answer to your question	0.32%
Hi	0.30%
Swiss Branded Watches	0.30%
Customer Receipt/Purchase Confirmation	0.29%
E-mail Handling Opinion Needed	0.29%
Replica Watches	0.28%
You've received a greeting ecard	0.22%
Return mail	0.21%
Great Finds	0.18%
Exquisite Replica	0.17%

Table 22: Most Popular Spam Subject Lines, 2009

Figure 8. Most popular spam subject lines, 2009

NETWORK SECURITY

Figure 9 and Figure 10 show the geographic distribution of spam URLs in 2009.

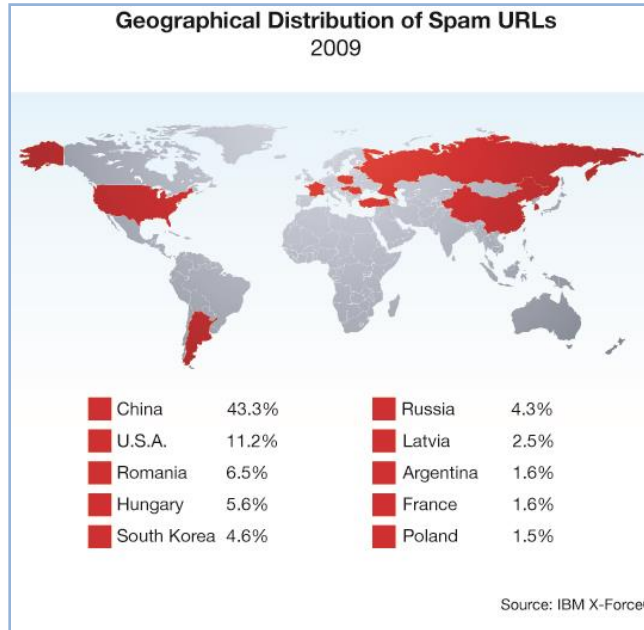


Figure 9. Geographic distribution of spam URLs

Country	2009 Volume	2008 Volume	Volume Change
Brazil	7.7	2.7	192.6%
United States	6.6	8.3	-20.3%
India	3.6	1.6	130.4%
South Korea	3.1	1.7	81.2%
Turkey	2.6	3.8	-31.3%
Vietnam	2.5	0.5	367.7%
China	2.4	3.2	-24.3%
Poland	2.4	1.6	43.4%
Russia	2.3	3.7	-38.2%
Argentina	1.5	1.3	16.0%

Volume in trillions per year Source: Cisco Security Intelligence Operations

Figure 10. Source of spam, 2009

The amount of spam is expected to rise by 30–40% in 2010. Anti-spam vendors indicate that 90% of the spam is 'soft,' relatively easy to identify and filter. The remaining 10% is considered 'hard,' requiring 90% of their efforts to find and filter. The hard messages are highly targeted and customized—for high-valued targets. This type of spamming is sometimes called *spear-phishing* or *whaling*. Spam messages often use domain names that are very close to those of valid companies, for example, *pay-pal.com* versus *paypal.com*.

Malicious adware/scareware

These are scams that trick a user into downloading and executing software, which may or may not contain malware. For example, the screen shown in Figure 11 invites the user to download a free scan program, which actually is malware.



Figure 11. Example scareware advertisement

Adware and scareware is a rapidly growing industry, up by 585% in just the first half of 2009.

Making money from malware

Successful malware can result in a number of unpleasant effects:

- **Botnets:** Formed from infected computers under remote control. Botnets are used for generating spam and for distributed denial of service attacks.
- **Stolen data:** Eventually leading to stolen money, either through fraudulent credit card transactions or banking transfers.
- **Disabled or damaged computers:** Requiring significant amounts of time to restore or rebuild.
- **Partially or completely disabled services:** Such as e-mail or Web commerce.

Criminals are reaping benefits through the following ways:

- Unauthorized bank and credit card transactions.
Advance fees, as in the Nigerian scam requesting money to cover the transfer of millions of 'unclaimed' funds.
- Product sales from scareware and Web-based enticements.
Criminal services that allow the creation and use of malware, including the following:
Malware toolkits, as in the Zeus trojan toolkit.
- Resale of stolen credit card and bank account information.
- CAPTCHA-breaking services. CAPTCHA is a technique that presents an image with an embedded word or number, as shown in 0. This ensures that a human is involved in the

NETWORK SECURITY

interaction. Criminal elements are now offering software, services, and personnel to defeat this interaction.



CAPTCHA examples

- Virus testing services. These are online services that determine whether a candidate virus/malware file will be detected by 40 or more anti-virus programs.
- Search redirection. These are services that poison Google and other search engine lookups so that they direct users to target Web sites.
- Legal institutions may be perceived as insecure by their customers.

Network Security Testing

Network security is a critical concern for enterprises, government agencies, and organizations of all sizes. Today's advanced threats demand a methodical approach to network security. In many industries, enhanced security is not an option. U.S. federal regulations such as Sarbanes-Oxley, HIPAA, GLBA, and others require organizations such as financial institutions, health care providers, and federal agencies to implement stringent security programs to protect digital assets.

The layered approach represents the best practice for securing a network. It is based on maintaining appropriate security measures and procedures at five different levels within a network:

1. Perimeter
2. Network
3. Host
4. Application
5. Data

Network security professionals speak in terms of 'work factor,' which is an important concept when implementing layered security. Work factor is defined as the effort required by an intruder to compromise one or more security measures, which in turn allows the network to be successfully breached. A network with a high work factor is difficult to break into, while a network with a low work factor can be compromised relatively easily. If hackers determine that a network has a high work factor, which is a benefit of the layered approach, they are likely to move on and seek networks that are less secure.

Figure 12 details the accepted security levels, along with the types of security tools used at each level. Ixia tests products and software at the perimeter and network levels, which will be the subject of this document.

NETWORK SECURITY

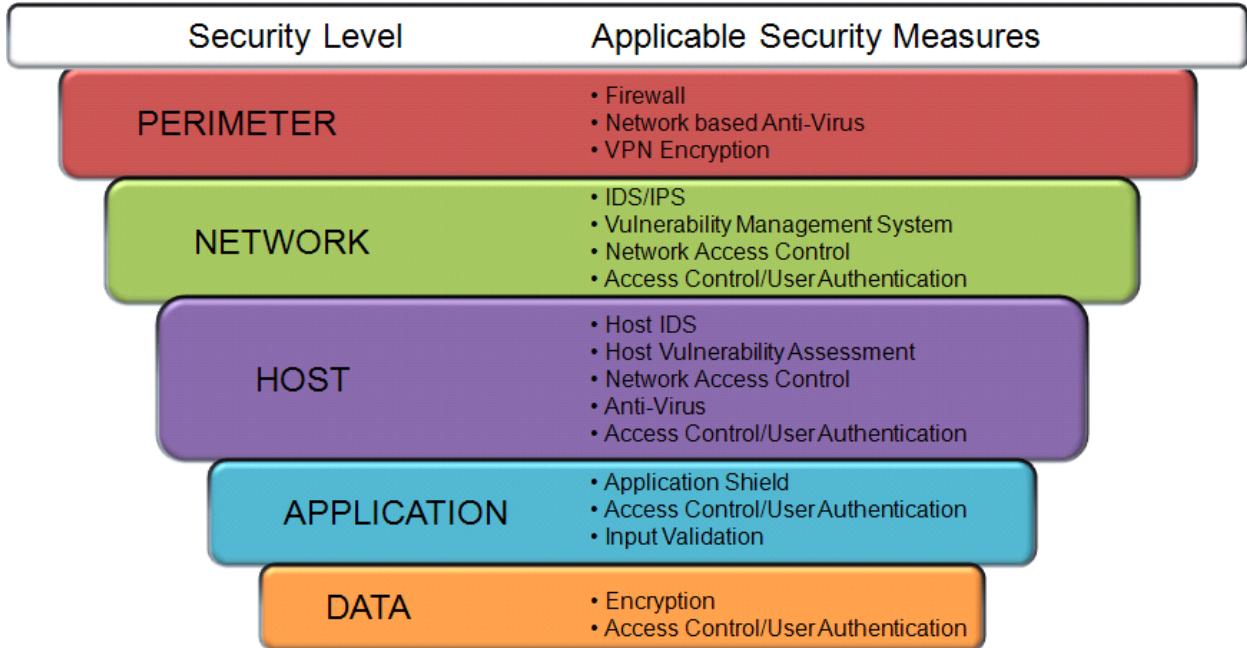


Figure 12. Security levels

NETWORK SECURITY

Network Security Devices

Figure 13 is a simplified diagram of an enterprise network, complete with security devices.

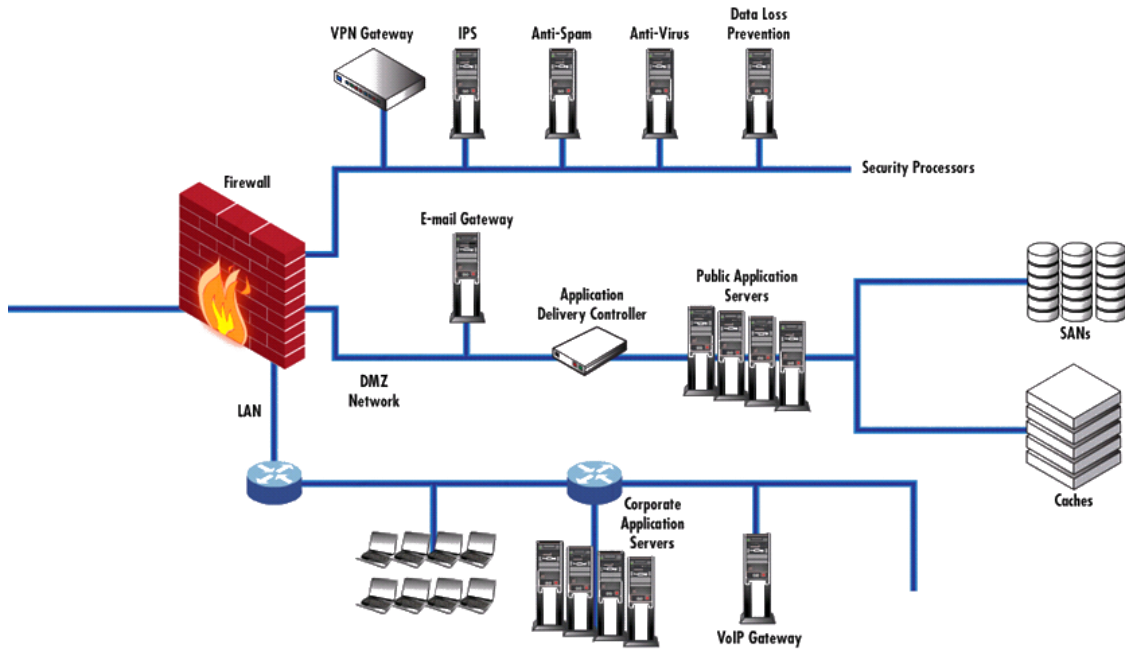


Figure 13. Simplified secured enterprise network

The security components that will be discussed in the following sections include:

- Firewall
- VPN gateway
- Intrusion detection and prevention systems (IDS/IPS)
- URL filtering
- Anti-virus
- Anti-spam
- Data loss/leakage prevention

The security processors, when they are not integrated into a unified threat management (UTM) device, are normally connected to a private network connected to the firewall. Servers that offer public services, such as e-mail and Web are kept on a private network called the demilitarized zone (DMZ). These private networks serve to isolate them from the local area network (LAN) users.

NETWORK SECURITY

Firewalls

Firewalls were the first independent security devices used with external network connections. The purpose of the original firewalls was to ensure that only required connections were allowed into the enterprise network. This typically includes services offered to the public: e-mail, Web, FTP, DNS, and a few others. Firewalls are also used to limit the types of services that internal computers may access outside the enterprise. This serves to somewhat limit malware from contacting external servers.

Firewalls initially operated by filtering connections based on a 5-tuple, as shown in Figure 14:

TCP or UDP

Source IP address

Source port number

Destination IP address

Destination port number

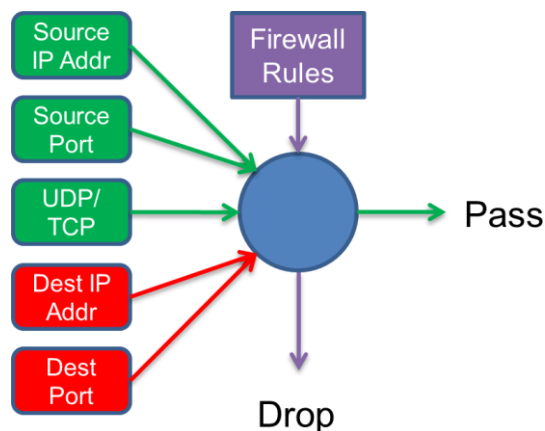


Figure 14. Basic firewall operation

Firewall rules are applied against connections attempted through the firewall, either inbound or outbound, to determine whether the connection is allowed or not. This worked well for a number of years, but as services and their protocols multiplied and applications began to use HTTP's port 80 as their transport mechanism, the ability of firewalls to meaningfully control traffic diminished.

NETWORK SECURITY

To handle this, firewalls began to use a technique, one of which is known as deep packet inspection (DPI), as shown in Figure 15.

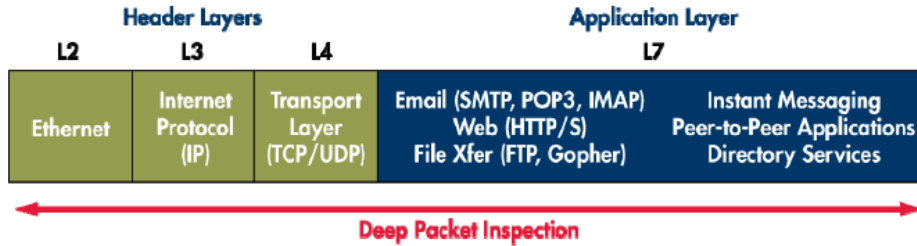


Figure 15. Deep packet inspection

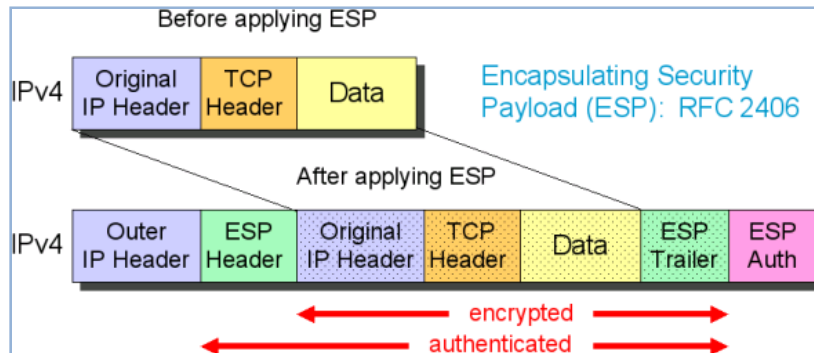
In addition to using the 5-tuple information included in layers 2, 3, and 4 of a packet, DPI looks into layer 7 application information to determine exactly the service that is being used. This additional information is then used in firewall rules.

VPN gateway

VPN gateways are used to securely connect multiple sites within an enterprise, remote and roaming employees, and business partners. Two protocols are commonly used:

SSL. This protects and encrypts traffic, while providing a Web-based interface for information access.

IPsec. This is network-level security that encapsulates and encrypts all traffic between the gateways, as shown. IPsec is described in detail in the **Error! Reference source not found.** *VPN Test Methodologies* section.



IPsec encapsulation

The original packet is encapsulated within a new packet that includes an additional encapsulated security payload (ESP) header. The header and additional trailers (and an optional authentication header (AH)) serve to ensure that the source of the packet can be validated.

IPsec is used when multiple sites wish full, transparent access to each other's networks.

Intrusion Detection and Prevention Systems (IDS/IPS)

Intrusion detection systems are an older technology that passively monitors network traffic, looking for particular malicious patterns, such as repeated attempts to log on to an account. When they notice a pattern, they send alerts to administrators and sometimes modify firewall rules to restrict access from the offending IP address.

Intrusion prevention systems are logically in line with traffic. That is, all traffic from the firewall's external link is sent through the IPS. It is responsible for identifying and stopping suspected traffic. Specific IPS rules and signatures are used to control how many flows are watched and for how long so as to ensure that the IPS does not significantly diminish the overall traffic flow. IPSs are complex systems, attempting to minimize the number of false positives.

URL Filtering

URL filtering seeks to keep users away from a restricted set of Web sites. These sites are generally classified as follows:

- Offensive content: pornography or other objectionable material.
- Harmful content: containing malicious code.
- Inappropriate content: pages deemed not proper to view at work, such as games or sports.

The list of Web sites used with the first two categories is often distributed as a service from a security vendor, based on the experience of all of its customers. IT managers create and maintain the last category, often based on lists from the security vendor.

Anti-Virus

Network anti-virus software, located on the firewall or UTM system, serves to identify and filter all forms of malware. It does this by looking at the network connections associated with protected services: e-mail, Web, FTP, IM, and others. The data within the stream is examined using a number of techniques that identify malware. Depending on the particular software, the connection or transfer may be aborted or the offending malware removed from the stream.

Each vendor has a set of proprietary techniques that they use to identify malware. A common technique is the use of signatures, which are particular unique sequences or bits of data that identify the malware.

Anti-Spam

Anti-spam network software has a great deal in common with anti-virus software, and is often bundled together. Spam is a growing problem, with more and more sophisticated, customized messages being delivered. List-based approaches often miss such messages. Users must remain skeptical and vigilant with respect to 'special' offers.

Data Loss and Leakage Prevention

Data loss/leakage prevention (DLP) is different than other security precautions in that it looks at outbound versus inbound information. DLP seeks to keep company and client proprietary information from leaving the organization, either innocently or maliciously.

Outbound information flows, such as e-mail, Web form data, FTP, IM, and other channels are filtered. A list of rules, keywords, and policies are applied to determine whether the communication should be rejected or allowed. Such filtering is very tricky. For example, a brokerage company might disallow any account number to be sent to a customer, who may be frustrating for the broker and customer.

Evasion Techniques

Security devices have a tough job—operating on large traffic volumes and keeping up with an ever changing set of threats.

An additional complication is the ability of hackers to disguise their attack through evasion techniques. A few examples are as follows:

- **URL obfuscation.** URLs filtering may be confused by the use of backslashes instead of forward slashes, or the use of % escape characters instead of 'normal' letters.
- **Fragmentation.** IP packets are broken up into many smaller pieces, making it more difficult to identify.
- **Stream segmentation.** An attack taking place over one connection, e-mail for example, might be interspersed with other traffic, potentially over a long period of time. Security appliances may need to stop looking at the original connection for lack of space.

Testing Security Devices

Testing of network security devices requires a number of techniques, which will be discussed in the next few sections:

- Known vulnerabilities
- Data loss tests
- Massive denial of service
- Protocol robustness
- Realistic multiplay traffic with comprehensive quality of service metrics
- Encrypted traffic

Known Vulnerability Testing

Known vulnerability testing is the cornerstone of network security device testing. Attacks are mounted against the security device by using a large database of known malware, intrusions, and other attacks. A number of organizations exist to maintain this list. One leading organization is the U.S. National Vulnerability Database maintained by the National Institute of Standards

NETWORK SECURITY

and Technology (NIST). The Mitre Corporation provides access to this database, called the CVE—Common Vulnerabilities and Exposures. As of May 2010, more than 42,000 vulnerabilities are listed, with more than 15 added on a daily basis.

Proper security testing requires that a number of known vulnerabilities be applied to security devices at a significant percentage of line rate. The device under test (DUT) should properly reject all such attacks, while maintaining a reasonable rate of transmission of 'good' communications.

In addition, known vulnerabilities must be applied using the wide variety of evasion techniques. The combination of thousands of known vulnerabilities and dozens of evasion techniques requires that a subset of all possibilities be used for testing. Test tools offer representative samples, including special cases for newly published vulnerabilities.

Data Leakage Testing

Data leakage testing involves transmission of data from the 'inside-out' to determine if data loss prevention devices will detect the leakage of proscribed information. All outbound means must be tested, including e-mail, e-mail attachments, Web-based mail, Web form data, FTP, and IM.

Enterprises must create test cases for each of the rules, keywords, and policies that they use in the security device, including tests that should not be flagged. Network equipment manufacturers (NEMs) have a more difficult job—requiring a more extensive set of test cases that exercise each type of rule and policy, along with a sampling of keywords.

Distributed Denial of Service

Denial of service attacks often use large numbers of computers that have been taken over by hackers. Those computers use dozens of attack techniques designed to overload network and security devices. This type of testing requires test equipment capable of simulating thousands of computers.

The DUT must be tested to ensure that none of the denial of service attacks, singly or in combination, is able to disable the device. In addition, the ability of the DUT to accept new connections and provide an acceptable level of performance must be measured.

Protocol Robustness

There are literally hundreds of protocols associated with modern Internet systems. Each operating system vendor and network equipment manufacturer implements each protocol in its own way. Many protocols are used in the deployment of end-user services. For example, Figure 16 shows some of the protocols used in a voice over IP (VoIP) call.

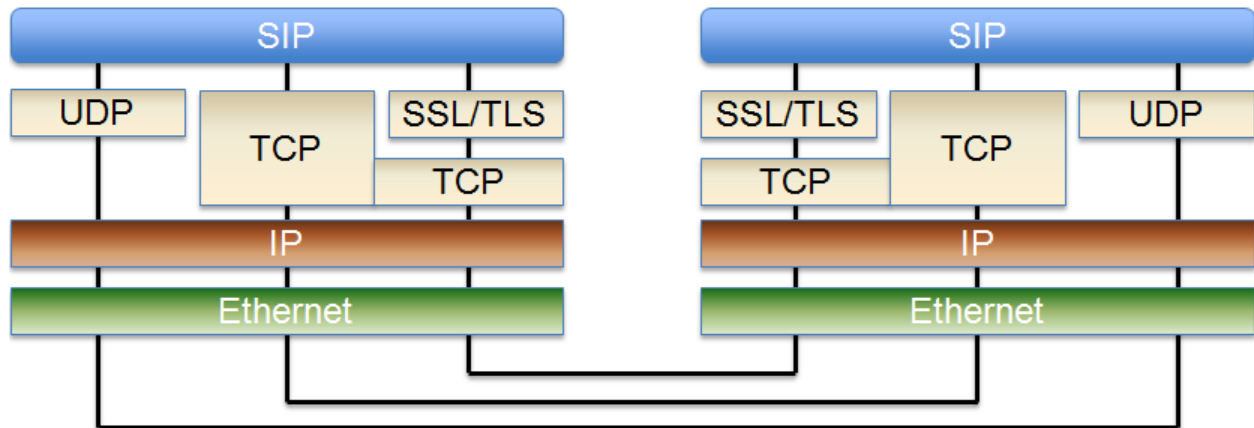


Figure 16. VoIP Protocols

Each and every protocol implements a complex state machine, complete with multiple state transitions and options handling. A perfect protocol implementation will properly handle each legal and illegal input.

These protocol implementations are generally tested for conformance to standards and proper functionality, but seldom extensively tested. Extensive testing requires that all corners of the protocols' implementation be tested. This type of testing is called protocol robustness/resilience testing, measuring the ability of a network device to handle unusual and malicious input. This type of testing finds design, configuration, and implementation flaws.

These types of flaws are often called 'zero-day' flaws, because they remain undiscovered until the first day of their deployment. These flaws can be particularly expensive; a newly offered service cannot be removed without loss of reputation or revenue. Some flaws have remained unpatched for as long as two years.

In principle, such testing could be accomplished by long sequences of random input, but the sophistication of today's protocols would require too long a period of time. The technique used for this type of testing is referred to as intelligent 'fuzzing.'

Intelligent fuzzing has an understanding of a protocol's state machine and the fields in the protocol that represent options. The fuzzing test machine uses this understanding to exercise a protocol's state machine, taking it through all normal legal transitions while trying illegal inputs along the way. In addition, all plausible options are attempted along the way. During testing, fuzzing checks for proper protocol behavior by monitoring the network connection.

Line-Rate Multiplay Traffic

Not only must security devices fend off attacks, but they must pass non-malicious traffic at the same time. To ensure this, it is necessary to test for defense against attacks with a background of real-world multiplay traffic. That is, a mix of voice, video, data, and other services that constitute normal traffic should be applied to the DUT such that the sum of the malicious and normal traffic is the maximum for the device's interfaces.

The quality of experience for each of the normal services must be measured to ensure that the end users' satisfaction will not be sacrificed. For example, voice over IP requires very little bandwidth, but latency and jitter impairments are immediately heard by the human ear.

Encrypted Traffic

As enterprises move to connect their multiple sites and mobile and remote users together into a corporate VPN, data encryption is becoming increasingly important. Data encryption ensures both privacy and authentication of the sending party through the use of certificates or other techniques.

The process of establishing an encrypted link, and then subsequent encryption and decryption can be a significant load for a security device. It is essential that a realistic mix of encrypted traffic be mixed with clear traffic during performance testing. The rate at which encrypted connections can be established is particularly important, representing how quickly a network can resume normal operation after an outage.

Test Methodologies for Known Vulnerabilities

Vulnerabilities represent flaws in a product that may allow malicious users to take control over the victim's computer, compromise data on it, allow remote execution of malicious code, or gain unauthorized access.

The test methodologies covered by this section is trying to address the effectiveness, accuracy, and performance impact of devices that can protect the network against known, published vulnerabilities. Such devices include Intrusion Prevention and Detection systems, Unified Threat Management systems, and new generation firewalls.

The key metrics that needs to be addressed while testing security devices include the following:

- Security effectiveness
- Resistance to evasion
- Detection accuracy
- Performance

Security Effectiveness

Security effectiveness refers to the ability of a network security device to detect and prevent threats. As detailed in the introductory section of the book, threats can consist in known vulnerabilities, unknown vulnerabilities, DoS and DDoS attacks, and malware.

Effectiveness measurements shall target the following:

- Effectiveness based on the security policy used
- Effectiveness based on attack vector
- Effectiveness based on attack published date
- Effectiveness based on attack source
- Effectiveness based on threat type

Effectiveness based on Security Policy

Network security devices can be tuned to achieve maximum security protection, but usually, the elevated security comes at a cost by impacting the performance. To achieve the best balance between the security risk and the cost of security solution, measurements shall be conducted against different security policies. Many of today's security devices include a default security policy. Those should not be taken for granted, as the effectiveness of a default policy may be significantly different among products. Some vendors tuned their default policies to achieve the highest performance while reducing the protection level, while others provided highest

protection with the cost in performance. Additionally, each deployment environment may require custom policies. Therefore, understanding the associated cost for a given security policy is important.

Effectiveness by attack vector

The largest number of known vulnerabilities target software that is used by a large number of users. Popular vendors like Microsoft, Adobe, Apple, and their applications are the main targets because they own large market share.

Assessing the effectiveness of the device should be determined in relation with the attack vector that exploits vulnerabilities specific to the environment where the IPS/IDS/UTM device is deployed.

The attack vector can be a set of vulnerabilities related to a vendor (for example, Microsoft, Apple, Adobe), or specific to an application (for example, Microsoft Internet Explorer, Mozilla Firefox).

Effectiveness by vulnerability's published date

New vulnerabilities are disclosed daily, while software vendors address many of them in new versions of their software. Regardless of the availability of a patch, vulnerable software continues to be used. The older attacks can still be effective if they are targeting the right vulnerable software. Therefore, they continue to be relevant and protection against them shall be considered.

Effectiveness by attack source (Internal Attacks vs. External Attacks)

Attacks can be classified as internal or external based on the source of the attack.

The most common type of attacks is the one initiated by an external attacker. Those attacks usually starts by scanning the network perimeter of the victim, understanding the open ports and applications and operating systems used. After vulnerable software is found, the attacker executes the attack against the application and operating system. By exploiting the vulnerability, the attacker can execute the code remotely on the victim's computer. The attacker can also get root access, thereby gaining full control over the victim's computer. In this type of attack, the attacker controls when the attack is initiated.

The internal attacks are targeting client-based vulnerabilities. As an example, an internal attack can be launched by a user who visits links that may exploit vulnerabilities in the browser, or open documents that are specially crafted to exploit vulnerabilities of application opening the document. This type of attacks is also referred as *target initiated* attacks because it relies on the victim's computer to initiate the attack. The time when the attack is initiated cannot be controlled by the attacker.

Resistance to Evasion

Security devices have a tough job—operating on large traffic volumes and keeping up with an ever changing set of threats. An additional complication is the ability of hackers to disguise their attack through evasion techniques.

Evasion techniques can be divided in several classes, including the following:

- **IP Fragmentation**
- **Stream segmentation.** An attack taking place over one connection, e-mail for example, might be interspersed with other traffic, potentially over a long period of time. Security appliances may need to stop looking at the original connection for lack of space.
- **Remote Procedure Call Fragmentation**

Remote Procedure call (RPC) is a protocol that an application can use to request a service from a program running on a remote computer. There are two variants of RPC implementation: Sun's RPC, also known as ONC RPC and Microsoft's RPC, also known as DCE RPC.

RPC provides several features that can be used by attackers as an evasion technique: support for fragmentation at application level, several ways to represent same data, option to create multiple bindings with a single request and, context alteration.
- **URL obfuscation.** URLs filtering may be confused by the use of backslashes instead of forward slashes, or the use of % escape characters instead of 'normal' letters.

Evasion techniques provide simple mechanisms to transmit the same attack(s) in camouflaged ways to bypass the detection of security products. Some of the evasion classes such as IP Fragmentation, Stream Segmentation, and RPC Fragmentation can be applied across all the attacks. Missing the detection when one of those evasion techniques fail, gives attackers the opportunity to use the entire selection of exploits that they own. Evasion techniques play a critical role in understanding the security risks and they should be mandatory in evaluations of security effectiveness of IPS/IDS and UTM devices.

Detection Accuracy

Network security devices such as IPS and UTMs are placed inline to block internal and external attacks. To distinguish legitimate traffic from malicious traffic, such devices include complex techniques for traffic analysis and detection, which may include deep packet inspection, statistical analysis, fingerprinting, signature dictionaries, regular expressions, and partial document matching. By filtering all the incoming and outgoing network traffic, valid connections may end up being blocked by the device, causing a denial of service. Therefore, the strength of the detection engine directly correlates with the detection accuracy.

Testing for accuracy is critical in ensuring that a solution has no false positives or negatives.

Performance Impact

One of the most common effects when additional devices are placed inline is the increased latency. End to end latencies exceeding 150 ms will start affecting the quality of VoIP calls. Excessive network latency can also cause applications to spend a large amount of time waiting for responses from its remote peer, resulting in lower bandwidth usage. Different security policies impact differently the performance. Another variable is introduced by the type of traffic. Parsing SIP traffic compared with HTTP traffic may result in a larger processing effort, therefore impacting the performance differently. Lastly, the presence of malicious traffic results in additional processing operations that the device needs to take care of, potentially impacting the performance.

Benchmarking network security devices should start with baseline tests to assess the raw forwarding performance of the device. In those tests, all the security services must be disabled and the device must act as a simple forwarding element.

Test cases must cover raw performance for UDP and TCP protocols

- UDP - RFC 2544 Throughput Measurements
- TCP - Maximum Concurrent Connections
- TCP - Maximum Connections Rate
- TCP - Maximum Throughput

Step by step instructions covering those use cases are included in the **Application Delivery Black Book** (p/n: 915-2610).

In the second step, the same test cases are repeated, but this time the security policies are enabled on the device. For each security benchmarking, the test cases must be repeated.

Because IPS/IDS and UTM devices relies on deep packet inspection, the following set of DPI test cases should be covered:

- Max DPI Capacity and Performance with HTTP

TEST METHODOLOGIES FOR KNOWN VULNERABILITIES

- Maximum DPI Capacity and Performance with Multiplay

The 'Multiplay Traffic' should match as close as possible the traffic mix seen in the deployment network. The traffic characteristics are different and the DUT's performance can be impacted differently. Profiles covering traffic patterns inspected by the IPS/IDS/UTM devices deployed by universities, enterprises, service providers, financial, and government organizations are good examples.

Detailed description of those test cases is covered in the **Application Delivery Black Book** (p/n: 915-2610).

After the baseline performance numbers are established, the tests must be repeated in the presence of malicious traffic. An assessment of the security effectiveness must be conducted while generating traffic at different capacities. Recommended values include 25 percent, 50 percent, 75 percent, 90 percent, and 99 percent of the capacity determined by using the baseline test cases.

During performance benchmarking, the quality of experience must be closely monitored. IxLoad provides a comprehensive set of statistics that can help you qualify the Quality of Experience (QoE) as seen from a user's point of view. As an example, when VoIP traffic is used, Mean Opinion Score (MOS), PESQ (Perceptual Evaluation of Speech Quality), registration time, call setup time, call tear down time, RTP packet loss, RTP Jitter, and RTP One Way Delay are some of the key metrics that IxLoad can provide. The VoIP Black Book provides use cases focused on VoIP protocols covering the QoE in more detail.

Test tools like IxLoad that provide a statefull implementation of the L4-7 protocols are recommended as they can realistically emulate the user behavior when the network is experiencing delays or congestions, which can lead to more retransmissions and higher delays amplifying issues and resulting in lower QoE.

Test Case: Measuring the Security Effectiveness of Intrusion Prevention Systems

Overview

Network-based Intrusion Prevention Systems (IPS) is playing an essential role in any enterprise and datacenter security solutions. This test determines the security effectiveness of a Network-based Intrusion Prevention System against published vulnerabilities targeting both client and server applications. IxLoad-Attack's Published Vulnerabilities and Malware plugin will be used to replicate the communication between attackers and vulnerable targets.

To baseline the security effectiveness, we recommend that you send the attack probes sequentially, at lower rates without any additional traffic. While the presence of additional benign traffic may impact the security effectiveness, we recommend that this type of test be executed upon identifying the list of attacks that are successfully blocked by the IPS, and use attacks that have been previously detected and blocked to assess any impact that legitimate application traffic may add.

Objective

This test measures the security effectiveness of network-based IPS against attacks targeting published vulnerabilities on client and server applications. This test uses the predefined list of *All Vulnerabilities (CRITICAL)* as an example, but you can run the test using user-defined lists (custom list of attacks).

Setup

The setup requires at least two test ports – one acting as an initiator and the other as a responder. The initiator port corresponds to the Published Vulnerabilities and Malware (PVM) test activity that hosts the list of attacks to be executed.

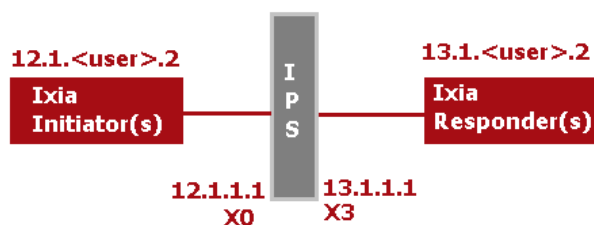


Figure 17. Test Setup

TEST CASE: MEASURING THE SECURITY EFFECTIVENESS OF INTRUSION PREVENTION SYSTEMS

Configure the policy of the device to allow both inbound and outbound communication for any traffic/protocol on any port (allow ANY to ANY). Configure the IPS to provide the maximum protection against exploits targeting published vulnerabilities.

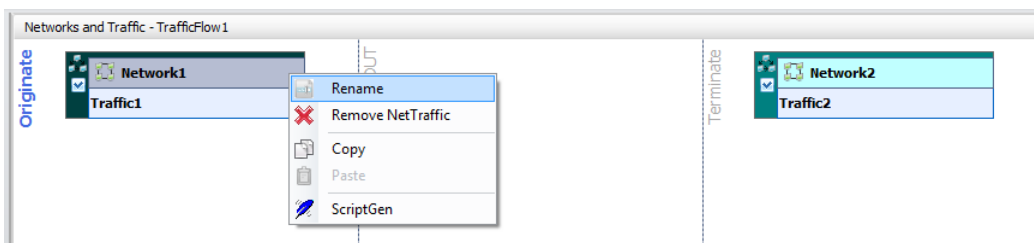
Step by Step Instructions

This section guides you through the IxLoad 6.0 configuration steps.


1. Define the Network and Traffic Flows

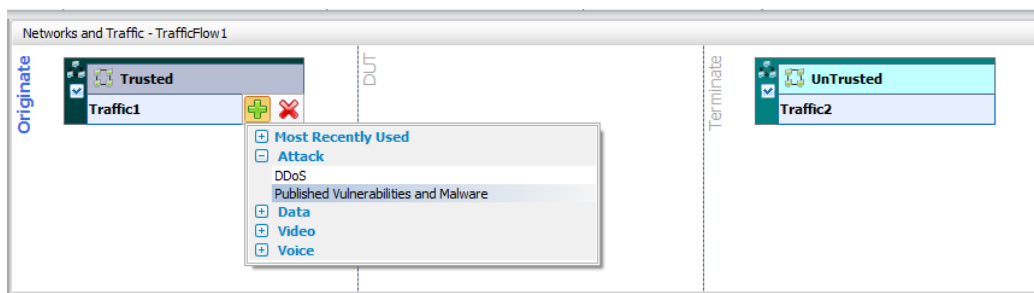
1.1. Create two networks, **Network1** and **Network2**.

- a. Rename¹ **Network1** to **Trusted**
- b. Rename **Network2** to **UnTrusted**



1.2. Add a **Published Vulnerabilities and Malware** activity to the **Trusted** network as follows:

- a. Position the mouse over the **Traffic1** object.
- b. Select the  button to display the traffic activities.
- c. Select the *Published Vulnerabilities and Malware* activity from the Attack group.



¹ To rename an object such as a Network, Traffic or Activity, select the object, *right-click* > *Rename*, then type the new name

TEST CASE: MEASURING THE SECURITY EFFECTIVENESS OF INTRUSION PREVENTION SYSTEMS

1.3. Similarly, add a **Published Vulnerabilities and Malware** activity to the **UnTrusted** network.



1.4. Rename **PublishedVulnerabil1** to **PVM_INIT**.

1.5. Rename **PublishedVulnerabil2** to **PVM_RESP**.



1.6. Set the following IP parameters for the **Trusted** network:

Network Name	IP Type	Address	Mask	Count	Gateway
Network1	IPv4	12.1.1.2	16	100	12.1.1.1

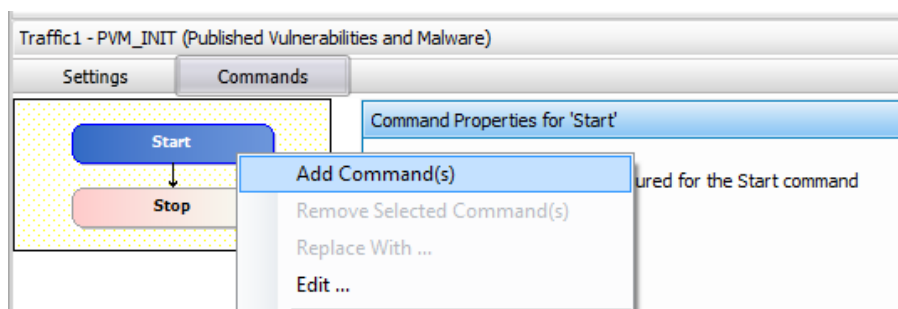
1.7. Set the following IP parameters for the **Untrusted**:

Network Name	IP Type	Address	Mask	Count	Gateway
Network2	IPv4	13.1.1.2	16	100	13.1.1.1

2. Configure the PVM_INIT activity

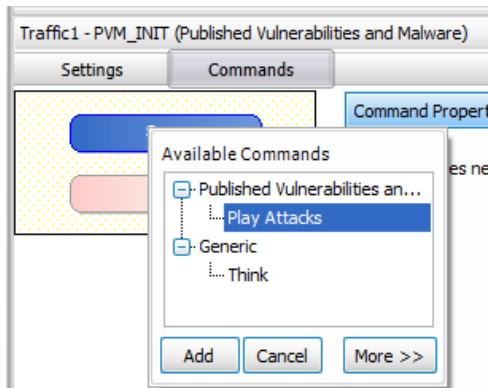
2.1. Add **Play Attacks** command to **PVM_INIT** activity as follows:

- Select the **PVM_INIT** traffic activity.
- Right-click the **START** command.
- Select the **Add Command(s)** entry.



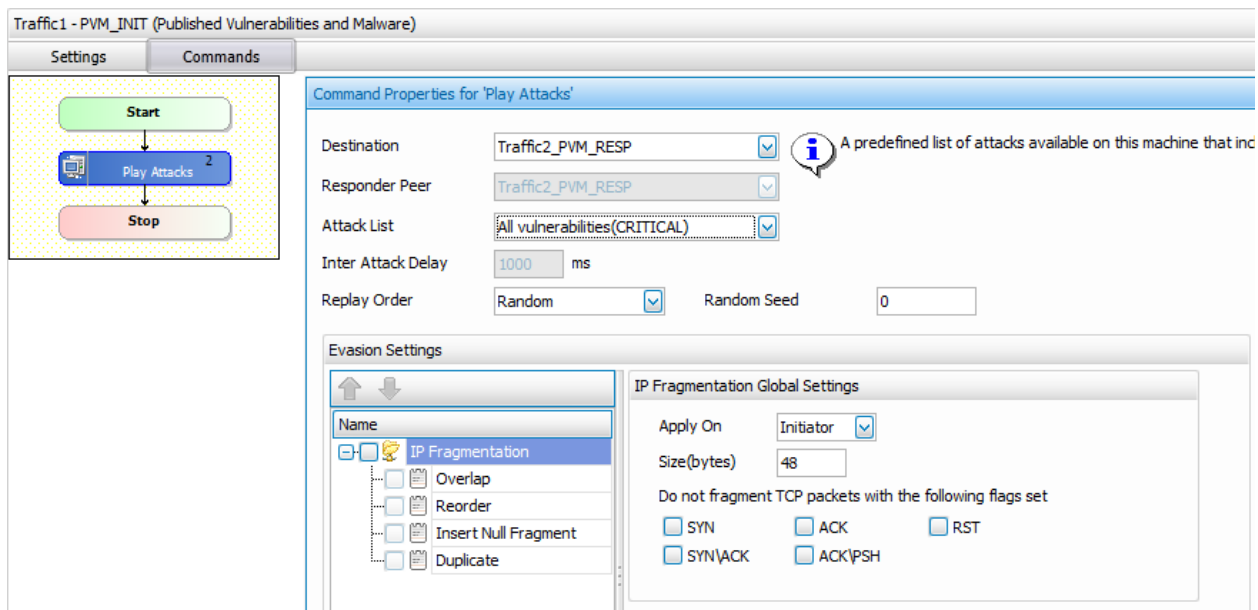
- Select the **Play Attacks** command; then click **Add**.

TEST CASE: MEASURING THE SECURITY EFFECTIVENESS OF INTRUSION PREVENTION SYSTEMS



2.2. Define the settings for the **Play Attacks** command as shown below:

- **Destination = Traffic2_PVM_RESP** (select from drop-down list)
- **Attack List = All Vulnerabilities (CRITICAL)**
Select <Create Attack List> entry to create your own list.
- **Replay Order = Random**

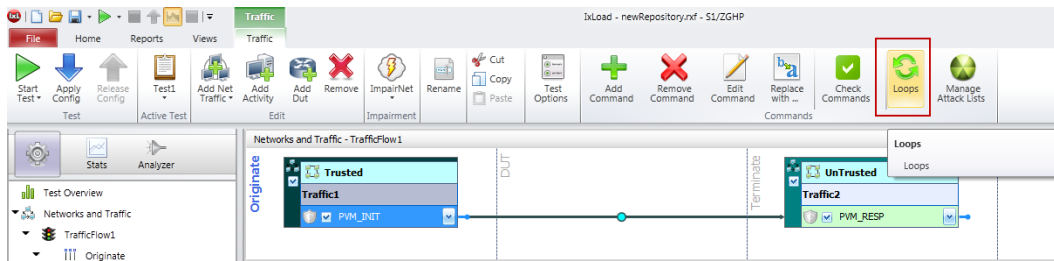


Notes:

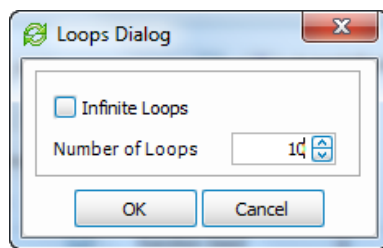
- since PLAY ATTACKS command is placed on a Trusted Network, all connections are initiated from the Trusted network
 - a faster alternative to steps 2.1 and 2.2 is to use the “*lollipop*” connector displayed on the right side of the **Initiators** activity. Drag a symbolic link to **Responders** (drag and release the mouse over the Responders activity). You will be prompted to select a predefined list of attacks. Select the *All Vulnerabilities (CRITICAL)* attack list.
- 2.3. By default the list of attacks is repeated infinitely. Change the loops count to a finite value (for example, Loop Count = 10)

TEST CASE: MEASURING THE SECURITY EFFECTIVENESS OF INTRUSION PREVENTION SYSTEMS

- a. Select the **PVM_INIT** traffic activity; the **Traffic** ribbon is displayed.
- b. Select the **Loops** button. The Loops Dialog box is displayed.

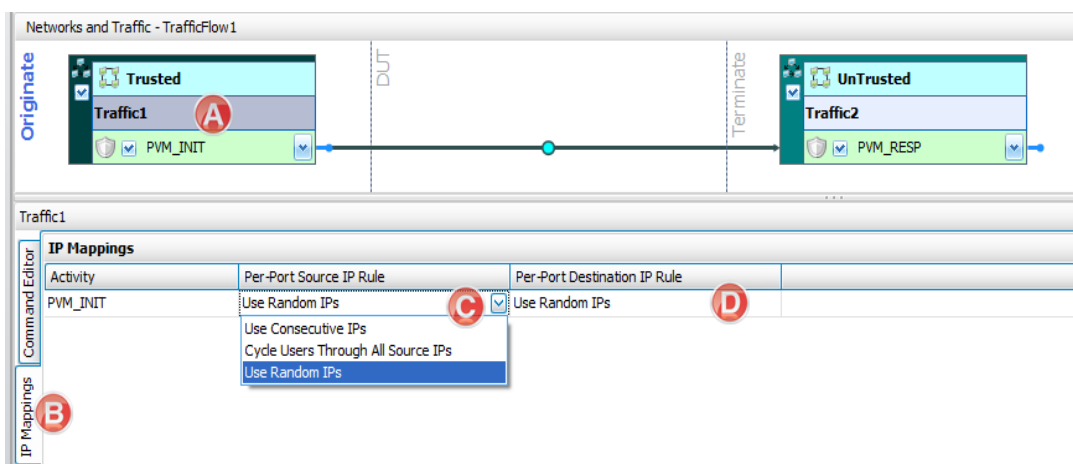


- c. Set the **Loop Count** to **10**
The value of 10 loop count is selected arbitrarily. To ensure consistency of the blocking, we recommend the use of multiple loops.



2.4. Enable IP randomization for both the source and the destination IP addresses as follows:

- a. Select **Traffic1** object.
- b. Select **IP Mappings** configuration page.
- c. Set **Per-Port Source IP Rule** to **Use Random IPs**.
- d. Set **Per-Port Destination IP Rule** to **Use Random IPs**.



3. Define the Test Objective details

3.1. In the Navigation pane, select **Timeline & Objective**.

TEST CASE: MEASURING THE SECURITY EFFECTIVENESS OF INTRUSION PREVENTION SYSTEMS

- 3.2. Set **Objective Type** as **Concurrent Attacks**.
- 3.3. Set **Objective Value** as **1** (max 100 per test port).
- 3.4. Keep the default **Ramp Up Type** as **Users/Interval**.
- 3.5. Keep the default **Ramp Up Value** to **1**.
- 3.6. Keep the default **Ramp Up Interval** to **1 second**.
- 3.7. Set **Sustain Time** to **1 hour**.
- 3.8. Keep **Ramp Down** value to 0 seconds.
- 3.9. Set **Ramp Down Time** to **10 seconds**.

The screenshot displays a software interface for configuring a test case. On the left is a navigation tree with 'Timeline and Objective' selected. The main window is titled 'Timeline and Objective' and contains a table of network traffic mappings and their objectives.

Network Traffic Mapping	Objective Type	Objective Value	% of Total Obj. Value	Timeline
TrafficFlow1				
Traffic1@Trusted	Concurrent Attacks	Total: 1	100.00	Timeline 1
P1M_INIT	Concurrent Attacks	1	100.00	Timeline 1
Traffic2@UnTrusted	N/A	N/A	N/A	<Match Longest>
P1M_RESP	N/A	N/A	N/A	<Match Longest>

Below the table, the 'Timeline' configuration is shown with the following settings:

- Ramp Up Type: Users/Interval
- Ramp Up Value: 1
- Ramp Up Interval: 0000:00:01
- Ramp Up Time: 0000:00:01
- Sustain Time: 0001:00:20
- Ramp Down Value: 0
- Ramp Down Time: 0000:00:20
- Iteration Time: 0001:00:41

The 'Iterations' section shows:

- Time to First Iteration: 0000:00:00
- Iterations: 1
- Time Between Iterations: N/A

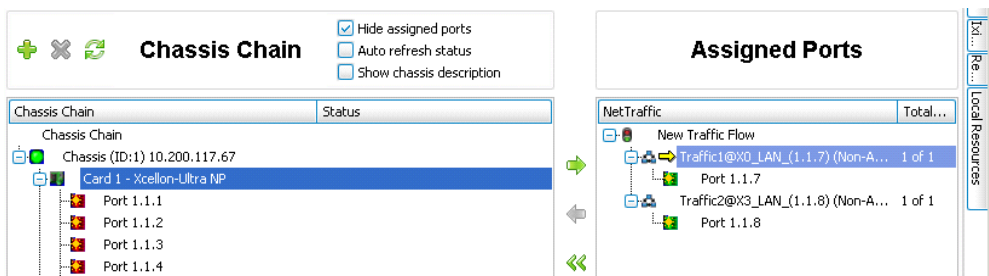
To the right of these settings is a timeline graph with a y-axis from 0 to 1 and an x-axis from 0:00:00 to 0:20:10. A red horizontal line is drawn at the value of 1, indicating the sustain time.

TEST CASE: MEASURING THE SECURITY EFFECTIVENESS OF INTRUSION PREVENTION SYSTEMS

4. Assign the Test Ports

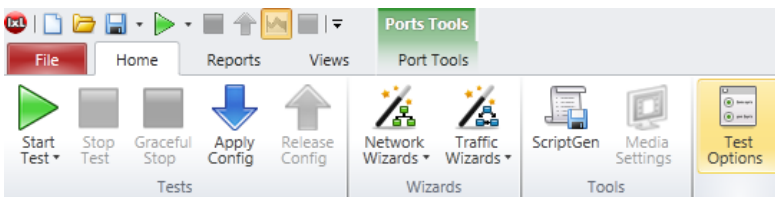
The test setup requires two test ports.

- 4.1. In the **Navigation** pane, select **Ports**.
- 4.2. Click the **Add Chassis** button.
- 4.3. Type the IP address of the Ixia chassis.
- 4.4. Assign the LAN port to the **Trusted** network and the WAN port to the **UnTrusted** network.



5. Define the Test Options

- 5.1. Using the ribbon menu, select **Home > Test Options**.
- 5.2. **Forcefully Take Ownership**.
- 5.3. Set **Reboot Ports before Configuring**.
- 5.4. Set **Release Configuration after Test**.



6. Run the Test

- 6.1. Save your configuration file using **File > Save** or **File > Save As ...**

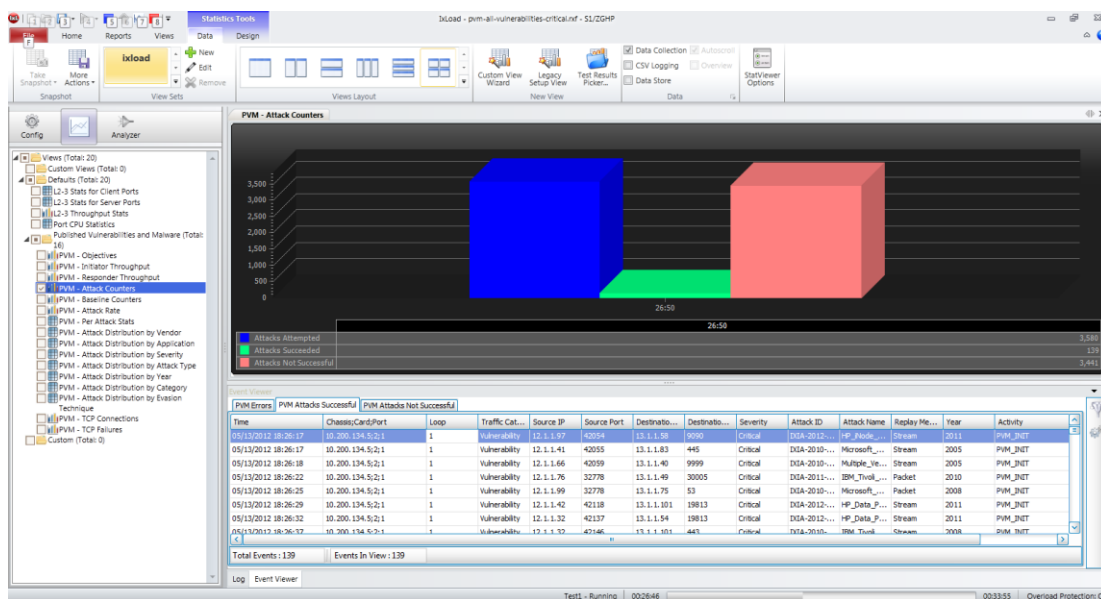
Example: **C:\VXIA\Test Cases\pvm-all-vulnerabilities-critical.rxf**

- 6.2. From the **Home** ribbon menu select **Start Test**

TEST CASE: MEASURING THE SECURITY EFFECTIVENESS OF INTRUSION PREVENTION SYSTEMS

Results Analysis

This section covers the key statistics and events that IxLoad provides for this type of test.



PVM Attack Counters

The PVM Attack Counters include three key metrics. The security effectiveness of the IPS can be calculated using the following formula:

$$\text{Security Effectiveness} = (\text{Attacks Not-Successful}) / (\text{Attacks Initiated}) * 100 [\%]$$

Metric	Description
Attacks Attempted	Counts the number of attack probes initiated by IxLoad.
Attacks Succeeded	Counts the number of attacks that are missed (have successfully traversed the Intrusion Prevention System).
Attacks Not Successful	Counts the number of attacks that are blocked (failed to successfully traverse the Intrusion Prevention System).

TEST CASE: MEASURING THE SECURITY EFFECTIVENESS OF INTRUSION PREVENTION SYSTEMS

PVM Events

IxLoad tracks the state and the key details corresponding to each attempted attack. By default, all the events are saved to two CSV files saved under results folder (for example: C:\Users\gzecheru\Documents\Ixia\IxLoad\6.0-EA\Results)

Loop	Traffic Cat...	Source IP	Source Port	Destinatio...	Destinatio...	Severity	Attack ID	Attack Name	Failure Reason	Failed
1	Vulnerability	12.1.1.81	42041	13.1.1.26	389	Critical	DXIA-2010-1107	Novell_eDirectory_LDAP_NULL_Search_Parameter_Buffer_Ov...	Connection dropped with TCP RST	7
1	Vulnerability	12.1.1.5	42042	13.1.1.35	80	Critical	DXIA-2010-1115	BEA_WebLogic_Server_Apache_Connector_HTTP_Version_St...	Connection dropped with TCP RST	4
1	Vulnerability	12.1.1.89	42043	13.1.1.38	7777	Critical	DXIA-2010-1046	HP_OpenView_Network_Node_Manager_ovw_dll_Message_H...	Connection dropped with TCP RST	4
1	Vulnerability	12.1.1.24	42044	13.1.1.3	7510	Critical	DXIA-2010-1034	HP_OpenView_Network_Node_Manager_HTTP_Handling_Buff...	Connection dropped with TCP RST	4
1	Vulnerability	12.1.1.74	42045	13.1.1.33	80	Critical	DXIA-2011-3803	IBM_Lotus_Domino_HPRAgentName_Parameter_Stack_Buffer...	Connection dropped with TCP RST	4
1	Vulnerability	12.1.1.27	42046	13.1.1.65	8080	Critical	DXIA-2011-3742	Apache_Struts2_ParametersInterceptor_Remote_Command_...	Connection dropped with TCP RST	4
1	Vulnerability	12.1.1.32	42047	13.1.1.49	1433	Critical	DXIA-2010-1101	Microsoft_SQL_Server_CONVERT_Function_Buffer_Overflow...	Connection dropped with TCP RST	4
1	Vulnerability	12.1.1.37	42048	13.1.1.98	3104	Critical	DXIA-2010-0882	CA_Products_Message_Queueing_Server_Buffer_Overflow_at...	Connection dropped with TCP RST	4
1	Vulnerability	12.1.1.48	42049	13.1.1.19	1500	Critical	DXIA-2010-1544	IBM_Tivoli_Storage_Manager_Express_Backup_Heap_Corrupt...	Connection dropped with TCP RST	8
1	Vulnerability	12.1.1.80	42050	13.1.1.73	8080	Critical	DXIA-2012-0092	Novell_Manager_getMultiPartParameters_Unauthorized_File...	Connection dropped with TCP RST	4
1	Vulnerability	12.1.1.41	42051	13.1.1.33	12401	Critical	DXIA-2011-3716	7T_Interactive_Graphical_SCADA_System_File_Operations_B...	Connection dropped with TCP RST	4
1	Vulnerability	12.1.1.35	42052	13.1.1.54	3465	Critical	DXIA-2010-0777	HP_OpenView_Client_Configuration_Manager_Radia_Notify_...	Connection dropped with TCP RST	4

PVM Attack Counters by Distribution Type

The IxLoad application allows you to list all the attacks for a given test, sorted based on:

- Attack Distribution by Year
- Attack Distribution by Severity
- Attack Distribution by Vendor
- Attack Distribution by Attack Type
- Attack Distribution by Attack Evasions
- Attack Distribution by Application

Stat Name	Attacks Attempted	Attacks Succeeded	Attacks Not Successful
1 10 Years Ago	0	0	0
2 2 Years Ago	480	30	450
3 3 Years Ago	340	0	340
4 4 Years Ago	490	40	450
5 5 Years Ago	570	10	560
6 6 Years Ago	360	0	360
7 7 Years Ago	430	30	400
8 8 Years Ago	250	0	250
9 9 Years Ago	0	0	0
10 Last Year	660	29	631
11 Other Years	0	0	0
12 This Year	0	0	0

Stat Name	Attacks Attempted	Attacks Succeeded
1 Others	1,490	
2 Microsoft	530	
3 Novell	380	
4 IBM	290	
5 HP	290	
6 Oracle	280	
7 Symantec	190	
8 Apple	100	
9 Computer Associates	30	
10 Mozilla	0	
11 Adobe	0	

Stat Name	Attacks Attempted	Attacks Succeeded
1 Malware	0	
2 Others	0	
3 Vulnerability	3,580	

TEST CASE: MEASURING THE SECURITY EFFECTIVENESS OF INTRUSION PREVENTION SYSTEMS

As a best practice, repeat the test several times and correlate the events with the logs provided by the Intrusion Prevention System.

4	05/13/2012 18:36:38.560	Alert	Intrusion Prevention	IPS Prevention Alert: BAD-FILES CUPS gif_read_lzw Function Buffer Overflow, SID: 4908, Priority: Medium	12.1.1.7, 44568, X2	13.1.1.90, 631, X3
5	05/13/2012 18:36:38.512	Alert	Intrusion Prevention	IPS Prevention Alert: EXPLOIT Server Application Shellcode Exploit 32, SID: 1114, Priority: Medium	12.1.1.34, 44567, X2	13.1.1.47, 8028, X3
6	05/13/2012 18:36:38.480	Alert	Intrusion Prevention	IPS Prevention Alert: DB-ATTACKS Ingres Database iidbms Heap Buffer Overflow 1, SID: 4759, Priority: Medium	12.1.1.50, 44566, X2	13.1.1.51, 48486, X3
7	05/13/2012 18:36:38.352	Alert	Intrusion Prevention	IPS Prevention Alert: DATABASE-APPS Microsoft SQL Server -- Connection Attempt 1, SID: 206, Priority: Low	12.1.1.28, 44565, X2	13.1.1.66, 1433, X3
8	05/13/2012 18:36:38.240	Alert	Intrusion Prevention	IPS Prevention Alert: PROTOCOLS CIFS -- IPC\$ Share Access, SID: 475, Priority: Low	12.1.1.57, 44564, X2	13.1.1.44, 139, X3
9	05/13/2012 18:36:38.128	Alert	Intrusion Prevention	IPS Prevention Alert: EXPLOIT HTTP Server Shellcode Exploit 7, SID: 6060, Priority: Medium	12.1.1.60, 44563, X2	13.1.1.91, 80, X3
10	05/13/2012 18:36:38.016	Alert	Intrusion Prevention	IPS Prevention Alert: SCADA-ATTACKS Measuresoft ScadaPro Remote Command Execution, SID: 6970, Priority: Medium	12.1.1.31, 44562, X2	13.1.1.13, 11234, X3
11	05/13/2012 18:36:37.848	Alert	Intrusion Prevention	IPS Prevention Alert: EXPLOIT Server Application Shellcode Exploit 28, SID: 5512, Priority: Medium	12.1.1.71, 44561, X2	13.1.1.40, 1500, X3
12	05/13/2012 18:36:37.816	Alert	Intrusion Prevention	IPS Prevention Alert: DOS Microsoft SMB2 Negotiate Request DoS (MS09-050), SID: 2032, Priority: Medium	12.1.1.8, 44560, X2	13.1.1.19, 445, X3
13	05/13/2012 18:36:37.768	Alert	Intrusion Prevention	IPS Prevention Alert: DATABASE-APPS GDS DB -- Connection Attempt, SID: 2319, Priority: Low	12.1.1.52, 44559, X2	13.1.1.80, 3050, X3

Test Variables

Test Tool Variables

Use the following test configuration parameters to repeat the test.

Test tool variables

Parameter Name	Current Value	Additional Options
IP Version	IPv4	IPv6
Test Objective	Concurrent Attacks (1)	Concurrent Attacks with up to 100 concurrent attacks per port Initiator Peer Count Test Objective
Attack Playlist	All Vulnerabilities (CRITICAL)	User Defined
IP Mapping	Random (source & destination IP addresses)	Use consecutive IPs (source, destination)
Benign Traffic	None	Add benign traffic to stress the CPU and/or memory utilization of the IPS.
Evasion Techniques	Disabled	IP Fragmentation (Run test with attacks that have been previously blocked/detected by the IPS)

Conclusions

This configuration covered the main parameters of the Published Vulnerabilities and Malware activity using a practical example allowing the user to baseline the security effectiveness of an Intrusion Prevention System.

Test Methodologies for DoS and DDoS

Denial of Service (DoS) and Distributed Denial of Service (DDoS) attacks are the oldest methods of attacking IP networks. While those methods are well-known and have been studied for years, they continue to remain one of the most effective ways to impact the performance of IP networks or services, or completely restrict access to a network, service, or application for legitimate users.

DoS versus DDoS

By definition, the intent of a DoS/DDoS attack is to partially restrict or completely deny access of legitimate users to resources provided by a victim's network, computer, or service. When this attempt is initiated from a single host, the attack is called a DoS attack. While some of the DoS attacks can be successful by using a single host with limited resource—compared with the victim's computer—the majority of the attacks require a group of malicious hosts to flood the victim's network by generating an overwhelming amount of attack packets. This type of attack is called DDoS.

According to Internet World Stats², the worldwide Internet population at the end of 2009 exceeded 1.8 billion users. Many of the Internet users browse the Internet without appropriate security software, or by using operating systems and software that is not properly patched. Those users have their systems vulnerable, allowing attackers to use automated techniques to discover such systems and use known vulnerabilities to install DDoS tools on their system. Such infected computers are named Zombie computers. Through automation, attackers exploit a large number of vulnerable computers, infecting them with malware software that gives attackers control to those systems.

WORLD INTERNET USAGE AND POPULATION STATISTICS						
World Regions	Population (2009 Est.)	Internet Users Dec. 31, 2000	Internet Users Latest Data	Penetration (% Population)	Growth 2000-2009	Users % of Table
Africa	991,002,342	4,514,400	86,217,900	8.7 %	1,809.8 %	4.8 %
Asia	3,808,070,503	114,304,000	764,435,900	20.1 %	568.8 %	42.4 %
Europe	803,850,858	105,096,093	425,773,571	53.0 %	305.1 %	23.6 %
Middle East	202,687,005	3,284,800	58,309,546	28.8 %	1,675.1 %	3.2 %
North America	340,831,831	108,096,800	259,561,000	76.2 %	140.1 %	14.4 %
Latin America/Caribbean	586,662,468	18,068,919	186,922,050	31.9 %	934.5 %	10.4 %
Oceania / Australia	34,700,201	7,620,480	21,110,490	60.8 %	177.0 %	1.2 %
WORLD TOTAL	6,767,805,208	360,985,492	1,802,330,457	26.6 %	399.3 %	100.0 %

Figure 18. Internet Usage and World Population Statistics (December 31, 2009)

A zombie computer reports back to a Command & Control center (C&C) by attempting a login session. After they are logged on, they become a part of a botnet that allows the attacker to

² <http://www.internetworldstats.com/stats.htm>

TEST METHODOLOGIES FOR DOS AND DDOS

control them. The most common C&C servers are Internet Relay Chat (IRC) servers, but in some cases, they can be Web servers as well.

Relying on hundreds to thousands of infected computers that have been previously infected with worms or trojans that facilitate remote control for an attacker, large DDoS attacks can be coordinated. Larger botnets can exceed 100,000 zombie computers, which can generate aggregated traffic of 10 Gbps to 100 Gbps.

Based on McAfee's third quarter report³ in 2009, 13 million new zombies were created in Q3/2009 with nearly 40 million new zombies created in the first three quarters of 2009. That is an average of 148,000 new zombies created every day this year. Based on the report data, the zombie computers were primarily used to generate spam, but their purpose could be easily changed by the botnet controller to generate DDoS attacks.

To increase the effect of the attack, vulnerabilities are often used to get control of Web servers and install trojans or worms that add the server to the controlled botnet. Server machines give the advantage of having better computing resources and their bandwidth is usually higher. Additionally, the attack traffic will be generated from trusted IPs.

Two types of DDoS attacks can be differentiated based on botnet's structure:

- DDoS attacks: the typical DDoS attack
- DRDoS attacks: a Distributed Reflector DoS attack

Motivation for DoS/DDoS attacks

DoS attacks are illegal activities. Regardless, such attacks continue to be frequently seen. They exist because they are easy to implement and the attack source is difficult to detect. A large number of tools are available on the Internet. The most common ones include Tribe Flood Network (TFN) and its newer version TFN2K, Trinoo (Trin00), Stacheldraht, myServer, Mstream, Omega, Trinity, Plague, and Derivatives.

3 http://www.mcafee.com/us/local_content/reports/7315rpt_threat_1009.pdf

Revenue driven/Monetary Gain

Motivated by monetary gain, many attackers advertise their service for DDoS attacks using underground forums. An example in McAfee's Third Quarter Threats Report⁴ for 2009 shows how a DDoS service provider is advertising its service consisting in a large botnet of anywhere between 80,000 to 120,000 bots that can generate 10 to 100 Gbps for a fee starting with USD 200 a day.



Figure 19. Example as shown in McAfee's Third Quarter Threats Report

To prove the size of the botnet owned, demonstrations of DDoS attacks are done by DDoS service providers to attract their customers. In many cases, many victims are randomly picked for such demonstrations. Competitive situations where the buyer rents DDoS services to cause loss in competitor's sales or to affect their reputation may be a strong motivator.

Cases of extortion using DDoS were reported as well. According to Sophos⁵, in 2006, an arrested group of Russian cyber-criminals made USD 4 million from blackmailing online gambling and casino Web sites.

Payback/Revenge

⁴ http://www.mcafee.com/us/local_content/reports/7315rpt_threat_1009.pdf

⁵ <http://www.sophos.com/pressoffice/news/articles/2006/10/extort-ddos-blackmail.html>

TEST METHODOLOGIES FOR DOS AND DDOS

On October 26, 2007, a UK based company, MoneyExpert.com⁶, experienced a DDoS attack, which put their site down during the weekend. During the day of the attack, the Web site planned to launch their payment protection insurance (PPI) reclaiming campaign, a massive new campaign to help many people recover thousands of pounds back on miss-sold insurance on loans, credit cards, store cards, and mortgages. Suspicions that the attack was the payback of the banks suffering the potential loss remained unproved. Another plausible cause could be the legitimate attempt to use the Web site from the large number of people that could benefit from the reclaiming campaign.

Unexpected Peak Hours

DDoS attacks can be the result of an overwhelming number of legitimate users trying to access Web sites announcing hot news or events that interest millions of users in a short time interval. One of the most publicized examples is where Google⁷ mistook the millions of search queries for a distributed DoS attack. Google search volume index chart depicted next shows the peak time was at 15:00 PDT.

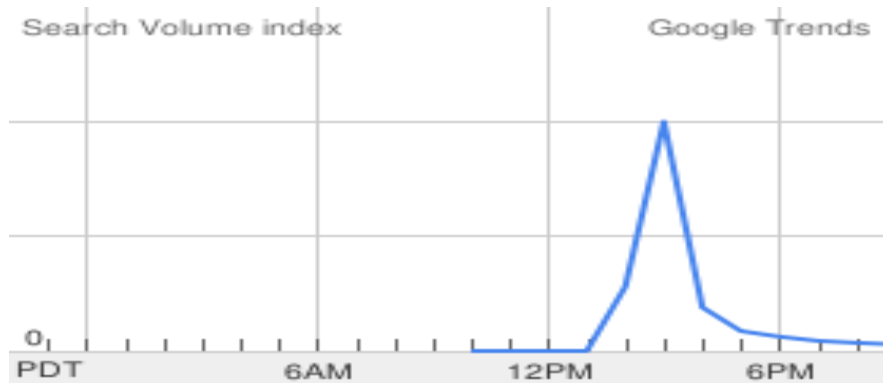


Figure 20. Google Trends - search for "Michael Jackson died"

⁶ <http://www.moneysavingexpert.com/site/moneysavingexpert.com-ddos-attack>

⁷ <http://www.christian-kalmar.com/google-michael-jacksons-ddos-attack/>

To filter legitimate traffic, Google prompted the users with an error page displaying a CAPTCHA field to let the users continue their query.

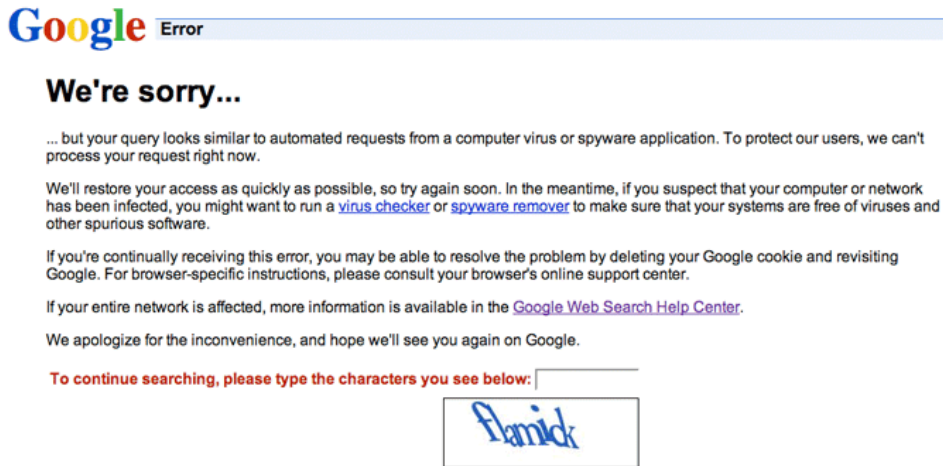


Figure 21. Google returned message to a valid search

Collateral damage

According to McAfee⁸, on August 6, 2009, a DDoS attack targeted social media sites hosting the account of a pro-Georgian blogger. As a result, the attack took down Twitter for several hours and significantly slowed down Facebook.

Miscellaneous

In many cases, the reason behind a DDoS attack remains unknown. Those attacks may simply have the intent of practicing an attack, or being initiated 'for fun.' Many video tutorials are available online accompanied by links to download the tools that can do the attacks.

⁸ <http://www.avertlabs.com/research/blog/index.php/2009/08/07/collateral-damage/>

DoS/DDoS: Methods of Attack

A large variety of DoS attacks can be attempted. Based on their intent, they can be classified as follows:

- **Resource starvation**
- **Alteration or destruction of system configurations**
- **Hardware damage**

The most common denial of service methods are based on overwhelming the victim's computer or network with useless data that result in overutilization of the following:

- **Network bandwidth**
- **CPU utilization**
- **Memory consumption**
- **Disk and storage**

Because of the limited nature of such resources on any system, they represent an easy and common target in DoS attacks. Usually, the attacks have a temporary effect and availability to resources is usually immediate after the DoS attack stops. Many cases were reported, however, where the victim was exposed continuously to DoS/DDoS for more than a week.

Based on the resources targeted, the attacks can be further classified as follows:

- **Bandwidth consumption**

One of the easier ways to deny access to a resource is done by consuming the bandwidth available between the ISP and the victim's network. Bandwidth can be easily consumed with any garbage data—UDP, ICMP, or TCP based traffic.

By consuming the entire bandwidth available, traffic from legitimate sources may result in connection drops. DoS attacks targeting bandwidth consumption requires a higher bandwidth than the victim's bandwidth or they are relying on amplification techniques. An example of DoS that uses amplification is the **Smurf attack**, which floods the victim's network by using spoofed ICMP messages sent to a broadcast address.

DDoS attacks are the most effective because the amplification relies on the high number of zombie computers generating the attack packets.

- **System resource starvation**

These attacks focus on consuming system resources such as CPU time and memory. CPU time is usually consumed with packets initiating new connections (for example, TCP SYN Flooding, HTTP GET Flooding, VoIP INVITE Flooding attacks) or packets targeting non-existing sessions (for example, TCP FIN Flooding, VoIP ACK flooding attacks).

Memory starvation can be achieved with legitimate connections that are maintained active after a connection is established.

By consuming these resources in an excessive manner, they become unavailable to legitimate users and systems.

- **DoS attacks targeting protocol and software flaws**

These attacks attempt to exploit design-flaws in software (for example, Ping of Death and Land attack). Those attacks do not require a large botnet to be effective; a single host machine can be used to send packets at low rates and lead to DoS by crashing the victim's computer, causing the computer to stop responding, or rebooting it.

Some examples of such DoS attacks that take advantage of the protocol's inherent design include SMURF, PING of Death, and Land Attacks.

- **Storage**

As a general rule, anything that allows data to be written to disk can be used to execute a DoS attack, assuming that no protection is set on the amount of data that can be written. As an example, an intruder may attempt to consume disk space by simulating actions that generates error messages on the victim's computer, errors that are logged and stored to the disk. Other examples may include massive amount of unsolicited e-mail messages or upload of useless data in unprotected locations (for example, network shares, ftp accounts).

- **Alteration or destruction of system configurations**

These types of attacks require access to the victim's computer. Exploits based on known vulnerabilities in the operating system or application itself may allow attackers to get root access to the system. By altering key configuration aspects of the server—routing tables, network configuration, user passwords, registry keys—or by destroying certain data (for example, the information stored in a database), an intruder may prevent users to access the compromised computer or network.

Routing-based DoS attacks target modification of routing table, preventing the victim to properly send or receive legitimate traffic.

To simplify the use of network addressing, name systems such as Domain Name Servers (DNS) provide a way to map the user-friendly name for a computer or service to the IP address associated with that name. DNS is a hierarchical naming

system and has the root domain on top of the hierarchy. An intruder that gains access to a DNS server can alter the cached data to direct legitimate traffic to wrong Internet (IP) addresses resulting in either flooding of a victim network or preventing the victim to send or receive any traffic.

- **Hardware damage**

Attackers that get root access to systems may destroy the hardware permanently. As an example, attempting to update the firmware of a device with a corrupted image may result in permanent damage.

Common DoS/DDoS Attacks

Address Resolution Protocol Flooding

Constantly sending Address Resolution Protocol (ARP) requests to the Gateway (or to another host within the same sub-network), thus tying up the attacked gateway or host. ARP attacks are confined to the sub-network in which the attacker resides.

ARP-based DDoS attacks require the attacker to have access to the victim's LAN. The attack is achieved by tricking the hosts of a LAN to generate a constant storm of ARP requests by providing them with wrong MAC addresses for hosts with already-known IP addresses. The victim can be either the local gateway or any host within the same sub-network. The large flood of ARP requests will lead to DoS.

TCP SYN Flooding

TCP SYN is one of the most common DDoS attack. A typical TCP connection requires a three-way handshake in which the client computer requests a new connection by sending a TCP SYN packet to its remote peer. In response, the TCP SYN/ACK packet is sent by the remote peer and the TCP connection request is placed to a queue, and continues to wait for the TCP ACK packet, which completes the handshake.

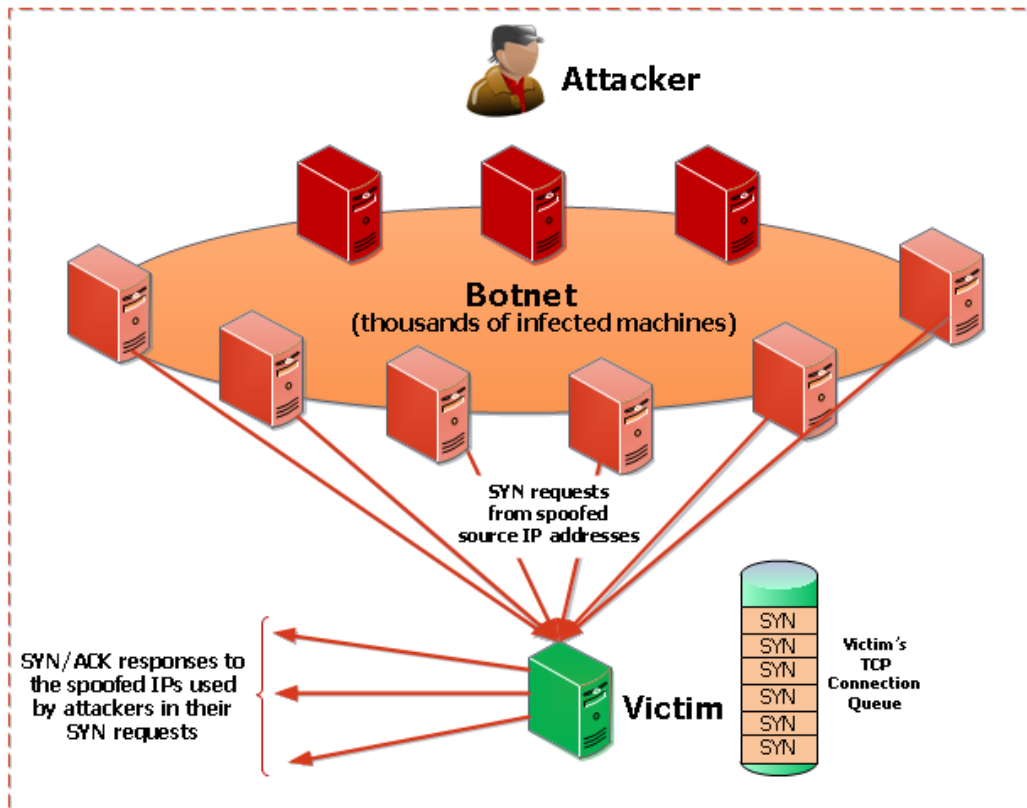


Figure 22. SYN Flooding DDoS attack

To achieve this attack, the attacker sends to victim's IP a storm of TCP SYN packets initiated from a large number of spoofed IPs forcing the victim to open a huge number of TCP connections and respond them with SYN/ACK. Because the attacker never ACKs the SYN/ACK packet, the victim will end up to a state in which it cannot accept new incoming TCP connections, regardless if they are coming from legitimate users.

UDP Flooding

A UDP flooding attack relies on a large number of attackers sending multiple UDP packets to the victim's computer, saturating its bandwidth with useless UDP packets. The attack packets can target open and closed ports. When the packets are targeting ports on which the victim's computer is not listening, ICMP Destination Unreachable packets may be replied by the victim to the spoofed IP included with each UDP packet. This will result in additional processing time and an additional storm of UDP packets destined to other computers.

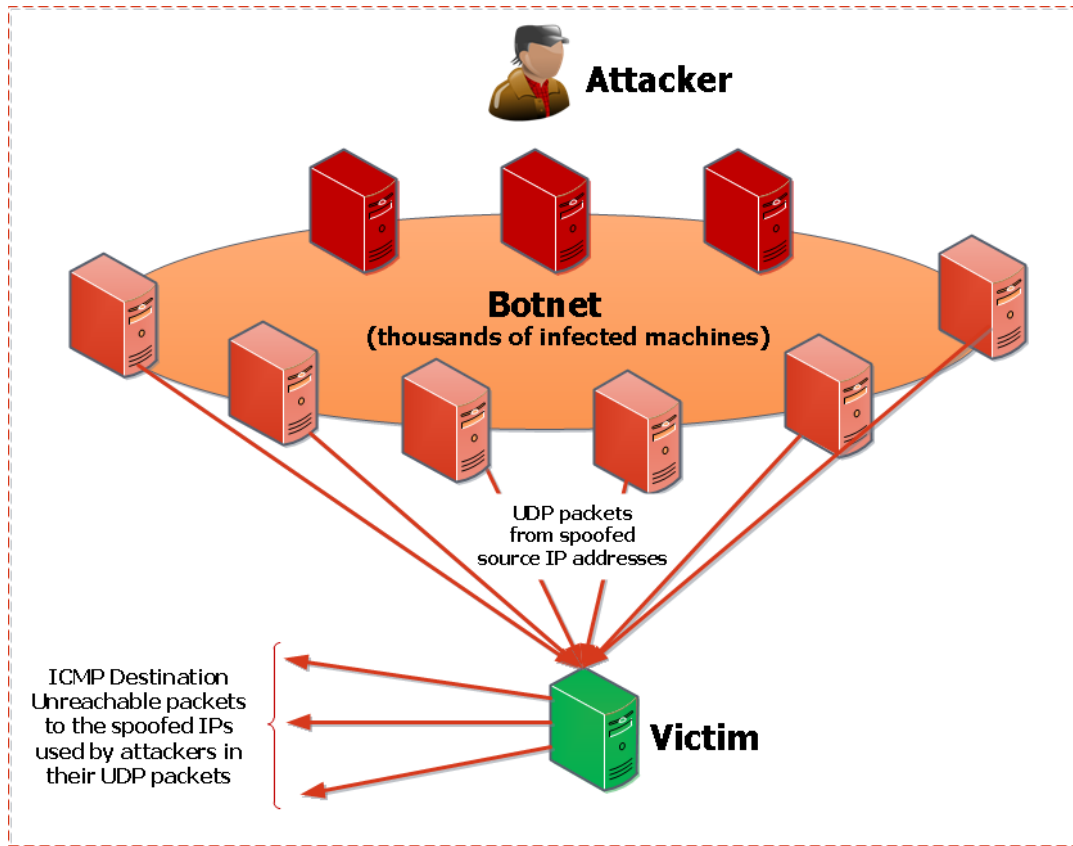


Figure 23. ICMP Flooding

PING Flooding attack

This threat floods the victim (a server or an end user) with multiple ICMP Echo request packets (PING), thus saturating its bandwidth. This is a very standard attack that can be done with utilities, such as PING, included with any operating system.

Smurf Attack

Smurf is yet another ICMP Echo request (PING) type of attack.

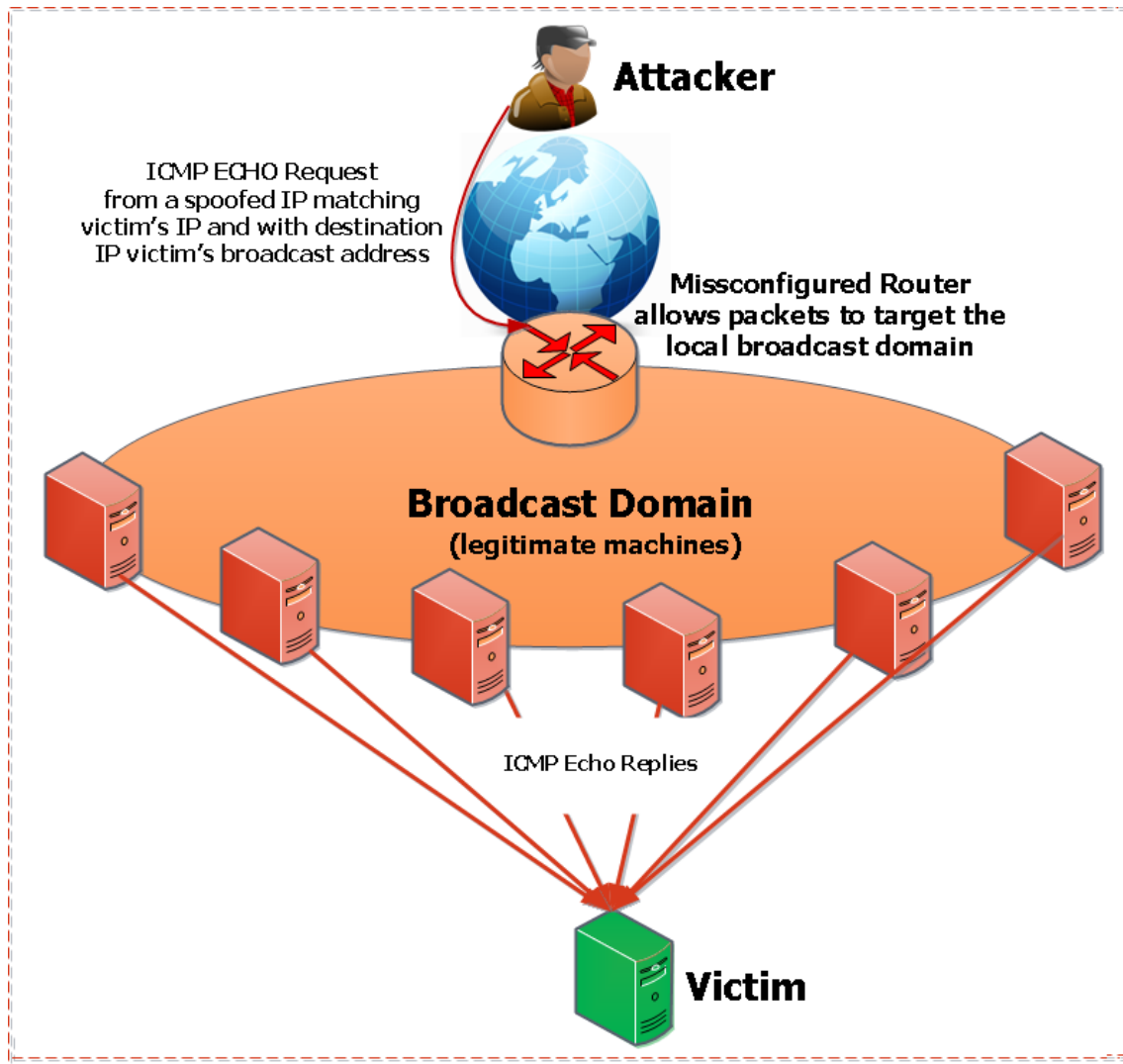


Figure 24. Smurf DoS attack

The attack exploits improperly configured networks, which allow external packets coming from the Internet to have the destination as an IP broadcast address. By sending a storm of ICMP Echo requests packets with the address spoofed with the intended victim's address, all the ICMP Echo requests are reflected back to all computes of the local network, resulting in an amplified number of replies destined to the victim's computer. The effect of this attack is the same with the PING Flooding attack.

Fraggle Attack

A Fraggle attack uses the same technique as described in a Smurf attack, except that the packets sent are UDP echo instead of ICMP Echo packets. The targeted services are *echo* (port 7) and *chargen* (port 19).

Defined by RFC 862, the ECHO service is an Internet protocol that listens on port 7 for either TCP or UDP port 7. On receipt of a packet, the ECHO protocol sends back a copy of data that it receives.

Defined by RFC 864, Character Generator (CHARGEN) is an Internet Protocol that listens on port 19 for UDP and TCP packets. On receipt of a UDP packet, a random number of characters are returned as a UDP response packet. On receipt of TCP packets, random characters are sent to the connecting host until the TCP connection is closed by the host.

PING of Death Attack

Similar to the Ping Attack, the Ping of Death also sends an ICMP Echo request to the victim. In this case, however, it is sent in the form of a fragmented message, which, when reassembled, is larger than the maximum legal size of 65,535 bytes. This might cause the attacked host to crash or to stop responding. A single Ping of Death Attack has the ability to incapacitate an unprotected victim for a fairly long period of time. Therefore, this type of attack may have a devastating effect even when sent at a very low rate.

ICMP 'Destination Unreachable'

On receipt of an ICMP 'Destination Unreachable' packet, the recipient will drop the corresponding connection immediately. This behavior can be exploited by an attacker by simply sending a forged ICMP Destination Unreachable packet to one of the legitimate communicating hosts. The DoS attack is achieved by breaking the communication of the legitimate hosts involved in communication.

ICMP 'Host Unreachable'

The ICMP Host Unreachable packet is another ICMP packet type that can be used to break the communication of two hosts. The DoS is achieved as described for ICMP 'Destination Unreachable', except that the packet type is ICMP 'Host Unreachable'.

ICMP 'Time Exceeded'

The Time Exceeded Message is an ICMP message that is generated by a gateway to inform the source of a discarded datagram because of the *time to live field* reaching zero. A time exceeded message may also be sent by a host if it fails to reassemble a fragmented datagram within its time limit.

This type of ICMP packet can also be used to break the communication of two hosts. The DoS is achieved as described for ICMP 'Destination Unreachable', except that the packet type is ICMP 'Time Exceeded'.

Land Attack

This attack attempts to 'drive the victim crazy' by sending it special-crafted TCP packets with the source IP address and source port number identical to the victim's IP address and port number. This causes the attacked host to think that it 'speaks to itself' and will often cause it to crash. This type of attack is ineffective against an updated system.

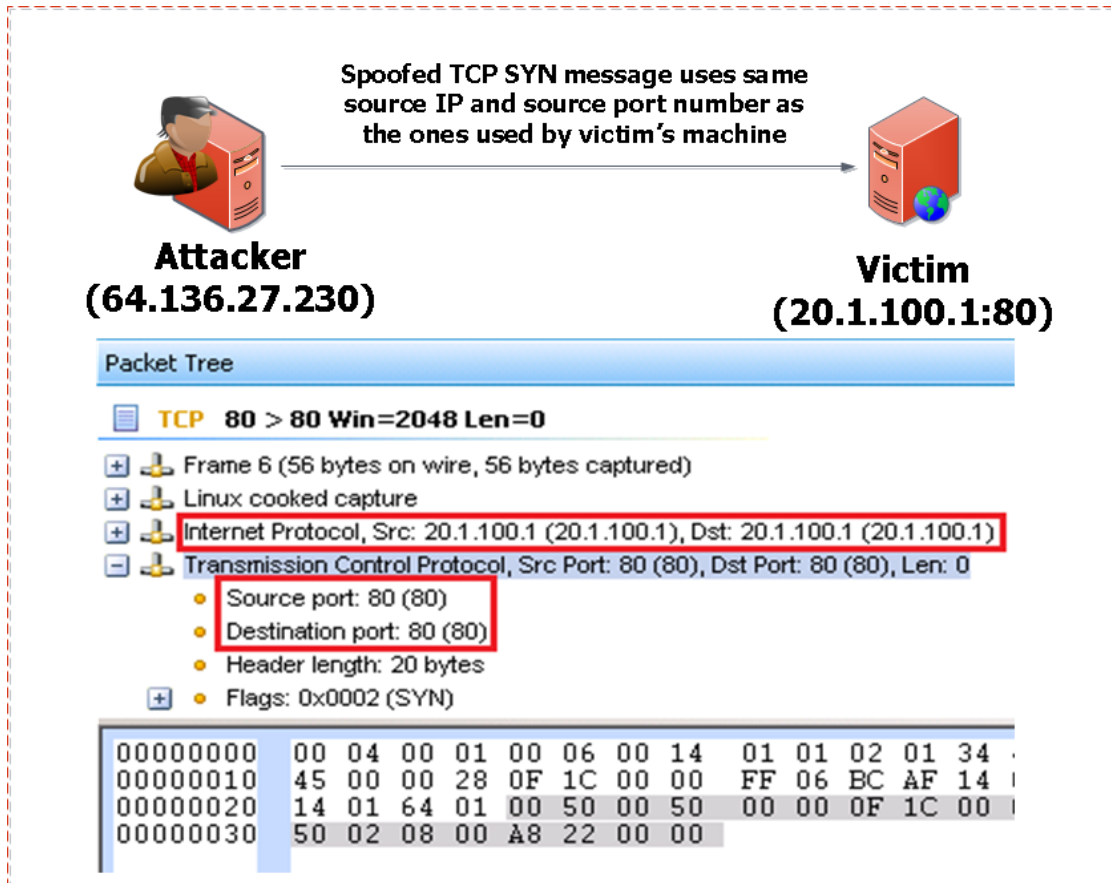


Figure 25. Sample crafted TCP SYN packet in a Land attack

The ICMP 'Redirect' message is commonly used by gateways when a host has mistakenly assumed that the destination is not on the local network. If an attacker forges an ICMP

'Redirect' message, it can cause another host to send packets for certain connections through the attacker's host.

Teardrop Attack

This is a fragmented message where the fragments overlap in a way that destroys the individual packet headers when the victim attempts to reconstruct the message. This may cause the victim to crash or stop responding.

FIN Flood Attack

This threat floods a user specified target with TCP packets from randomized, spoofed addresses, where the FIN (final) flag has been turned on. The FIN flag is sent by a user to designate that it is no longer sending packets. This attack is an attempt to flood the target with erroneous packets to hinder the performance and cause a slowed response to legitimate traffic and possibly DoS.

RST Attack

This vulnerability could allow an attacker to create a Denial of Service condition against existing TCP connections, resulting in premature session termination. Because an attack uses a random IP as the source IP, it is possible that the source IP or computer (if it exists) will send a reset packet (RST/ACK) back to the server that says it did not make the connection request. More likely, the IP address does not correspond to an active connection (because it is a random number); the server will keep trying to initiate a connection by resending SYN/ACK, and then RST/ACK (because it did not get any ACK back) packets back to the bogus source IP address. All this creates incomplete or half-open connections. RST attacks may cause route flapping (a router's continuous alternated advertising of destination networks through two different routes consuming that router's resources).

Application Level DoS and DDoS

Flaws in software implementations can be exploited to cause buffer overflow, consume all memory and CPU, crash the application stack, make the computer to stop responding, or reboot the computer.

Another group of DoS attacks rely on brute force, flooding the target with an overwhelming flux of packets depleting the target's system resources. Brute force attacks at application level floods the victim with the legitimate application requests that initiates transactions at application level. Examples of such attacks include HTTP GET/POST Flooding, SIP INVITE Flooding, DNS Flooding, and many others.

HTTP GET Flooding attack

The attack is achieved by sending an overwhelming number of HTTP GET or HTTP POST requests to the targeted HTTP Server, depleting the victim's resources. The requests have legitimate contents and they are originated over valid TCP connections. By serving those requests as normal requests, the server is ending up exhausting its resources.

By asking for large files stored on the server, the attack is amplified as a single legitimate request that can keep the server busy for a longer duration. A large HTTP GET Flooding attack was seen in U.S. and Korea in July 2009. The targets were Web sites of major organizations, news media, financial companies, and several government Web sites.

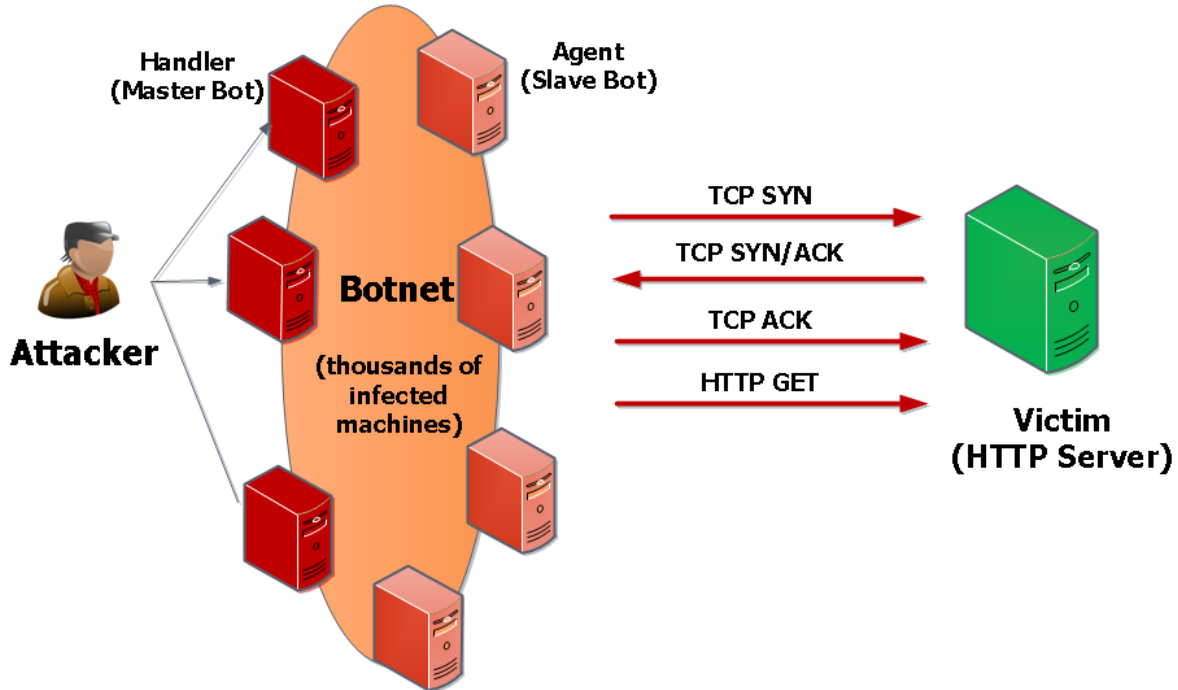


Figure 26. HTTP DDoS attack

DNS Flooding Attack

TEST METHODOLOGIES FOR DOS AND DDOS

This threat attacks a DNS server by sending a high number of DNS requests that look like they are initiated from the victim's IP address. The small queries sent by the zombie computers are amplified by the recursive DNS Servers that are used as intermediaries to resolve the domain, which generate in response to larger UDP packets, overwhelming the victim's computer. Another type of DNS Flooding attack can overwhelm a DNS Server by sending legitimate DNS queries to resolve random domain names, forcing the DNS Server to resolve them by initiating further queries to root servers and authoritative name servers. The storm of DNS queries may lead to resource depletion and therefore, causing the DoS effect.

SIP Flooding Attacks

This class of attacks floods the victim with a significant number of SIP messages, including REGISTER, INVITE, OPTIONS, MESSAGE, BYE, SUBSCRIBE, NOTIFY, ACK, and PING. The messages are sent from spoofed IP addresses and target depletion of victim's resources by forcing the victim to process useless SIP messages.

Test Case: Application Forwarding Performance under DoS Attacks

Overview

This test determines the degree of degradation that the denial of service (DoS) attacks have on a DUT's application forwarding performance.

Firewalls and DPI systems support advanced capabilities to protect it and the active user sessions from attacks. The security features of these devices add to the processing overhead of such devices and often come at the expense of impeding the overall performance of the system.

There are several approaches for testing the resiliency of a device under attack. This test focuses on determining the performance impact when the device under test is subjected to a network-based attack, such as a SYN Flood.

Objective

Determine the impact of network-based attacks on the performance of an application-aware device while processing and forwarding legitimate traffic.

Setup

The setup requires at least one server and one client port. In this test, the HTTP client traffic will pass through the DUT to reach the HTTP server. Next, dynamic DoS (DDoS) and malicious traffic will be introduced, with the appropriate inspection engines enabled on the DUT. To test realistic network conditions, several other legitimate protocols can be added.

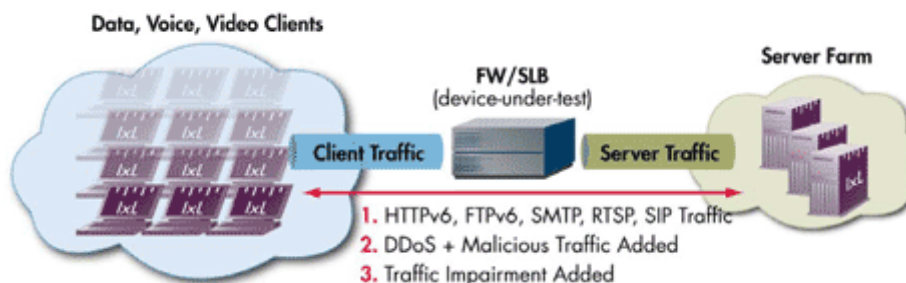


Figure 27. HTTP and DoS Attack Test Topology

Test Variables

Test Tool Variables

The following test configuration parameters provide the flexibility to create the traffic profile that a device would experience in a production network.

Test tool variables

Parameter	Description
Client network	100 IP addresses or more, use sequential or 'use all' IP addresses
HTTP client parameters	HTTP/1.1 without keep-alive 3 TCP connections per user 1 Transaction per TCP connection
TCP parameters	TCP RX and TX buffer at 4096 bytes
HTTP client command list	1 GET command – payload of 128 KB–1024 KB
HTTP servers	1 per Ixia test port, or more
HTTP server parameters	Random response delay – 0–20 ms Response timeout – 300 ms
DoS attacks	ARP flood attack, evasive UDP attack, land attack, ping of death attack, ping sweep attack, reset flood attack, smurf attack, SYN flood attack, TCP scan attack, tear-drop attack, UDP flood attack, UDP scan attack, unreachable host attack, and Xmas tree attack
Other protocols	FTP, SMTP, RTSP, SIP, or combination

TEST CASE: APPLICATION FORWARDING PERFORMANCE UNDER DOS ATTACKS

DUT Test Variables

There are several DUT scenarios. The following table outlines some of the capabilities of the DUT that can be switched on to run in a certain mode.

Sample DUT scenarios

Device(s)	Variation	Description
Server load balancer	Activate packet filtering rules	Configure SLB engine for 'stickiness' Change the algorithm for load balancing Use Bridged or Routed Mode for servers
Firewall	Activate access control rules	Configure Network Address Translation or Disable for Routed Mode
Firewall security device	Enable deep content inspection (DPI) rules	Advanced application-aware inspection engines enabled IDS or threat prevention mechanisms enabled

Step by Step Instructions

1. Configure the test to run a baseline test, which is an Ixia port-to-port test, to verify the test tool's performance.
2. Reference baseline performance: Fill in the following table with the baseline performance to use as a reference of the test tool's performance, based on the quantity and type of hardware available.

Reference baseline performance form

Performance indicator	Value per port pair	Load module type
Throughput		
Connections/sec		
Transactions/sec		

3. After you have obtained the baseline performance, set up the DUT and the test tool in accordance with the Setup section below. Refer to the Test and DUT Variables section for recommended configurations. Note that the network configurations must change between running the port-to-port and DUT test. Physical cabling will change to connect the test ports to the DUT. We recommend a layer 2 switch that has a high-performance backplane.
4. After you have the baseline, enable the security features of the DUT:
 - i. Enable application-aware inspection engines for virus, spam, and phishing attacks, which may be global parameters or access-lists.
 - ii. Enable application-gateway or proxy services for specific protocols used in the test, for example, SIP NAT traversal (STUN).
 - iii. Start IxLoad. In the main window, the **Scenario Editor** window is displayed. All test configurations will be performed here. To become familiar with the IxLoad GUI, see the Getting Started Guide section.
 - iv. Add the client NetTraffic object. Configure the client network with the total IP count, gateway, and VLAN, if used.
 - v. Add the server NetTraffic and configure the total number of servers that will be used.

For a step-by-step workflow, see Appendix A.

TEST CASE: APPLICATION FORWARDING PERFORMANCE UNDER DOS ATTACKS



Figure 28. IxLoad Scenario Editor view with client and server side NetTraffics and Activities

5. The TCP parameters that are used for a specific test type are important for optimizing the test tool. Refer to the Test Variables section to set the correct TCP parameters.

There are several other parameters that can be changed. Leave them at their default values unless you need to change them for testing requirements.

For a step-by-step workflow, see Appendix B.



Figure 29. TCP Buffer Settings Dialogue

6. Add the HTTP Server Activity to the server NetTraffic. Configure the HTTP Server Activity; the default values should be sufficient for this test.

For a step-by-step workflow, see Appendix C.

7. Add the HTTP Client Activity to the client NetTraffic. Configure the HTTP client with the parameters defined in the preceding Test Variables section.

You can use advanced network mapping capabilities to use sequence IPs or use all IPs configured.

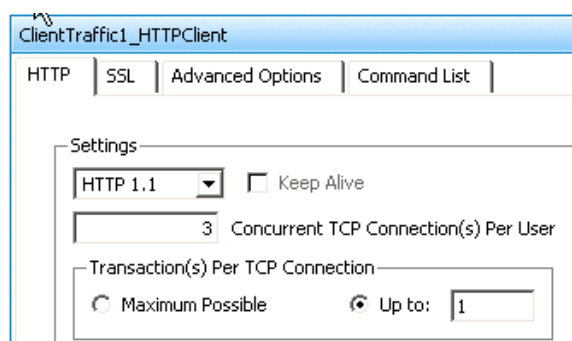


Figure 30. HTTP Client Protocol Settings Dialogue

For a step-by-step workflow, see Appendix D.

8. On the client traffic profile used to stress test the DUT, add a DoS Attacks Activity. Configure the DoS Attack client with the relevant DDoS attack signatures. You can optionally add other protocols to create a more stressful environment.

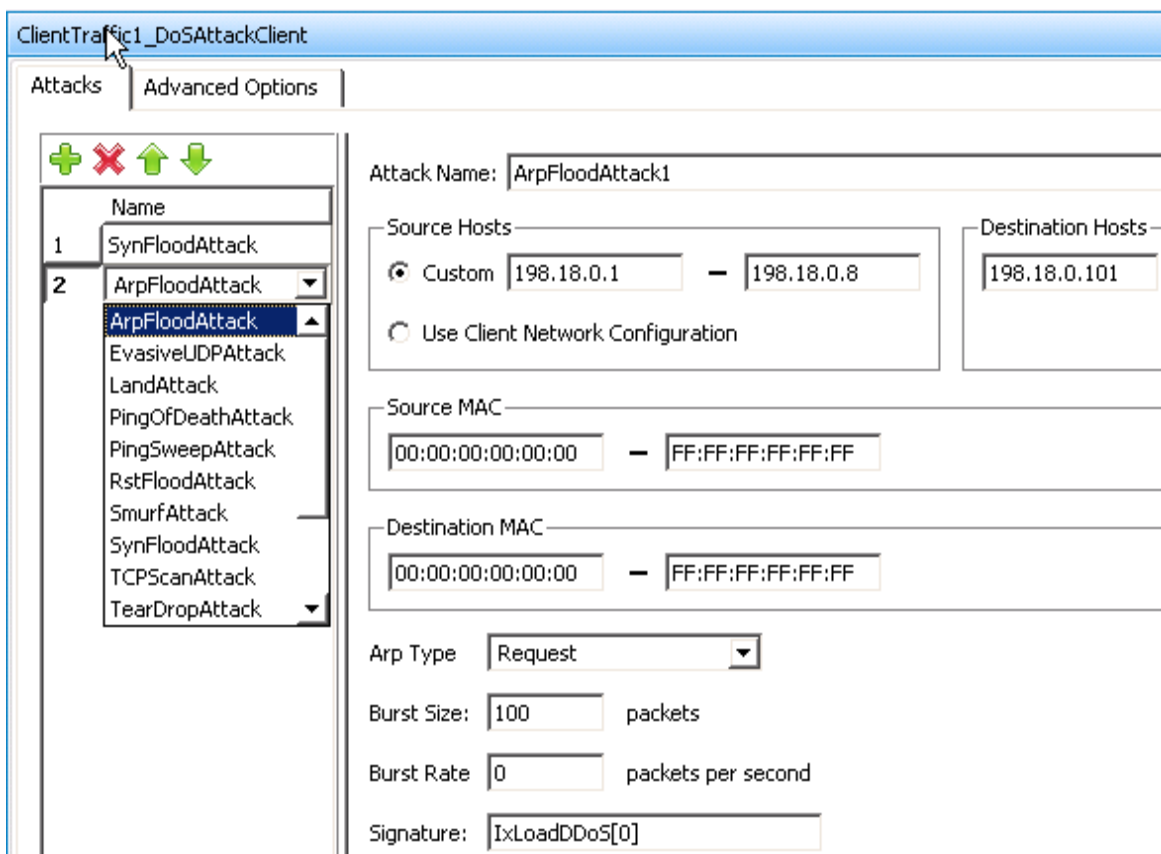


Figure 31. DoS Attack Client Settings Dialogue

If you want the attacks to originate from the same IP addresses as the legitimate HTTP traffic, select the **Use Client Network Configuration** check box. Alternatively, you can originate the attack from a different set of IP addresses.

There are several layer 7 DoS attacks to consider; you can add multiple attacks by clicking the + button. Use discretion in assembling the attacks to be initiated against the servers or DUT, and configure the Destination Hosts appropriately.

On the server side profile, add a PacketMonitorServer activity to monitor any attacks that were not discarded by a DUT, that is, attacks that make it through the DUT.

TEST CASE: APPLICATION FORWARDING PERFORMANCE UNDER DOS ATTACKS

To configure the PacketMonitorServer activity, simply select the corresponding DDoS Client activity. If the Automatic configuration mode is selected, the filters will be automatically imported from DoS attack configuration from the client network. Alternatively, you can use the Manual configuration mode to specify custom signatures.

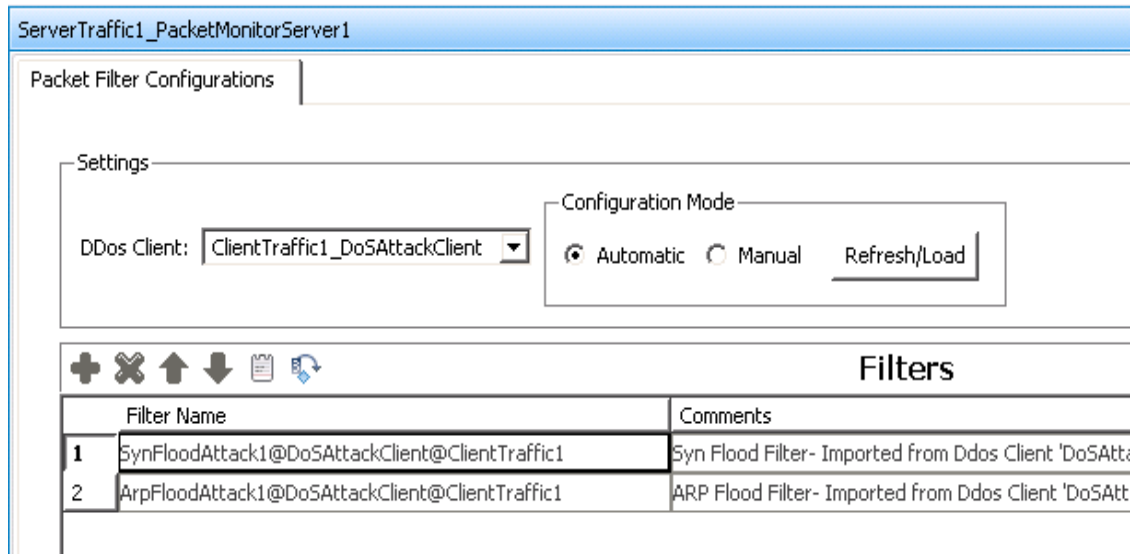


Figure 32. Packet Monitor Sever Settings Dialogue

Having set up client and server networks and the traffic profile, the test objective can now be configured.

9. Go to the Timeline and Objective view. The test objective can be applied on a per-activity or per-protocol basis. The iterative objectives will be set here and will be used between test runs to find the maximum TPS for the device.

Begin with setting the Throughput objective or one of the other metrics at the maximum achieved with no DoS attacks—this performance metric is the Reference baseline performance that was determined first.

For a step-by-step workflow, see Appendix E.

After the Test Objective is set, the Port CPU on the bottom indicates the total number of ports that are required.

TEST CASE: APPLICATION FORWARDING PERFORMANCE UNDER DOS ATTACKS

For the DoS Attack activity, set the objective to 1 simulated user and run the test. Use an iterative process to increase the simulated users to find the point at which the throughput or required objective starts to degrade.

Network Traffic Mapping	Objective Type	Objective Value
TrafficFlow1		
ClientTraffic1@ClientNetw...	Mixed Objective	Mixed Value
HTTPClient	Throughput (MBps)	113
DoSAttacks	Simulated Users	10

Figure 33. Test Objective Settings Dialogue

For a step-by-step workflow, see Appendix F.

Iterate through the test, setting different values for the Simulated Users objective for the DoS attack, which will gradually increase the intensity of the DoS attack directed at the DUT. Record the application throughput CPS, TPS, and throughput metrics for the test. Monitor the DUT for the target rate and any failure or error counters. Stop the iterative process when the DUT application forwarding performance drops below an acceptable level.

Results Analysis

To analyze the impact of DoS attacks on the DUT application forwarding performance, you need to compare the results from the performance baseline test case and results of the DoS attack test case. In addition, it is critical to analyze how various types of DoS attacks impact the performance.

The following are the key performance statistics that must be monitored. These statistics will help you identify if the device has reached its saturation point, and identify issues.

Key performance statistics to monitor

Metric	Key Performance Indicators	Statistics View
Performance Metrics	Connections/sec Total connections, Number of Simulated Users, Throughput	HTTP Client – Objectives HTTP Client – Throughput
Application Level Transactions	Requests Sent, Successful, Failed Request Aborted, Timeouts, Session Timeouts	HTTP Client – Transactions HTTP Client – HTTP Failures HTTP Client – Latencies
Application Level Failure Monitoring	Connect time	
TCP Connection Information	SYNs sent, SYN/SYN-ACKs Received	HTTP Client – TCP Connections HTTP Client – TCP Failures
TCP Failure Monitoring	RESET Sent, RESET Received, Retries, Timeouts	

TEST CASE: APPLICATION FORWARDING PERFORMANCE UNDER DOS ATTACKS

Metric	Key Performance Indicators	Statistics View
DoS Attacks	Successful, Failed Packets Bytes Sent	DDoS Client – Successful Packets DDoS Client – Failed Packets DDoS Client – Bytes Sent
Packet Monitor	Packets Received, Filtered and Allowed	Packet Monitor Server – Packet Statistics Total

Real-time Statistics

The following graph provides a view of the real-time statistics for the test. Real-time statistics provide instant access to key statistics that should be examined for failures at the TCP and HTTP protocol level.

In the following graph, you can see the throughput value was 410 Mbps before the DoS attacks began and how the throughput performance drops as the DoS attack intensity increases.

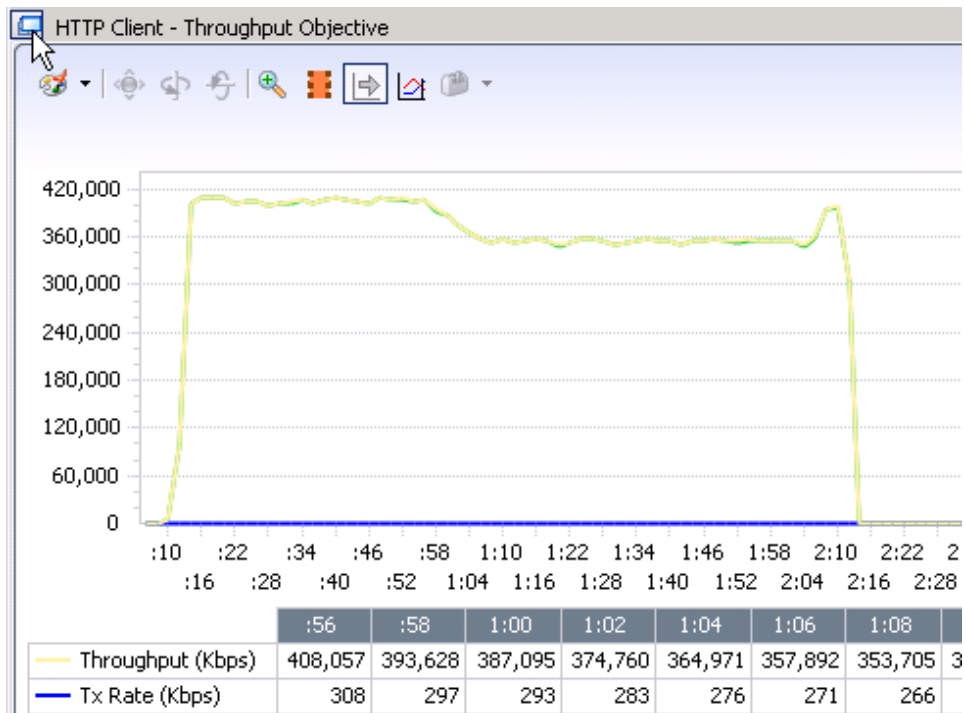


Figure 34. HTTP Throughput Statistics view

Note how at 0.58 sec the DoS attacks begin and the throughput starts to drop.

The following graph shows the corresponding DoS attack rate. You can see at the start of the test that there were no DoS attacks generated and during that period the throughput graph showed a steady 410 Mbps. When the DoS attacks started around 58 seconds, the throughput degradation began.

TEST CASE: APPLICATION FORWARDING PERFORMANCE UNDER DOS ATTACKS

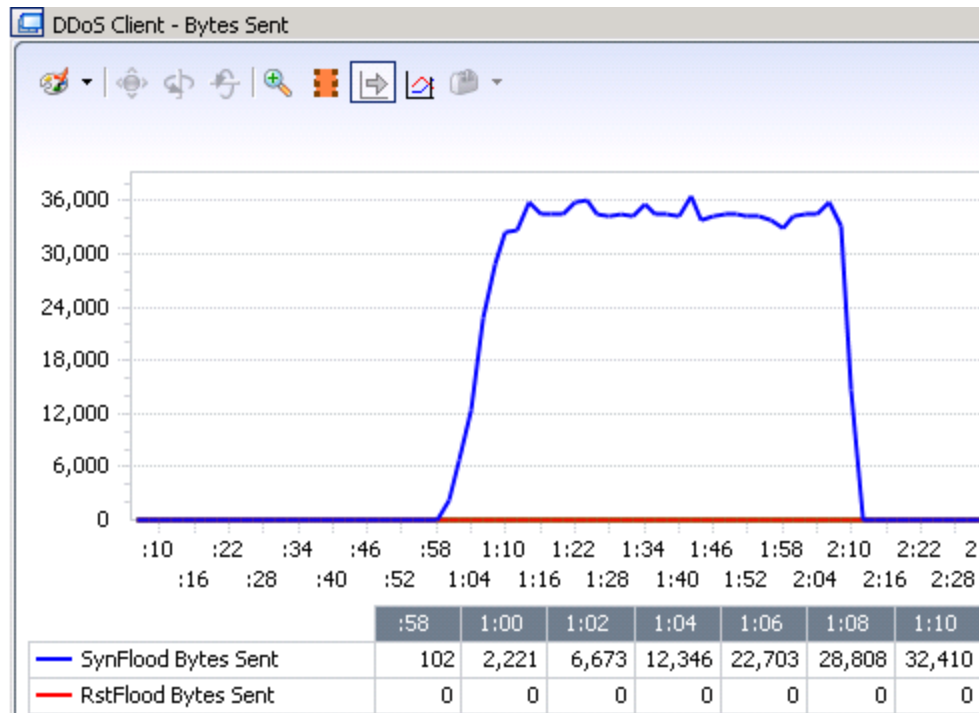


Figure 35. DDoS Client Bytes Sent Statistics View

Notice the time the SynFlood Attack began and the corresponding effect on the throughput graph.

Other metrics of interest are TCP and transactions failures. In some test runs, however, you may not see any TCP failures or transaction failures during a DoS attack. When you compare the total number of TCP connections serviced or total throughput during the DoS attack, as in the preceding case, you may notice degradation compared to the baseline test case values.

TEST CASE: APPLICATION FORWARDING PERFORMANCE UNDER DOS ATTACKS

A jump in latency is also observed during DoS attacks, as shown in the following figure. At the same time as the DoS attacks start, you see an increase in the HTTP time to last byte (TTLB) latency values. This indicates the device's inability to sufficiently transfer data at the inflection point when the attacks began.



Figure 36. HTTP Client Latency Statistics View

The preceding figure shows that the latency gets higher at the same time as the DoS attacks begin.

TEST CASE: APPLICATION FORWARDING PERFORMANCE UNDER DOS ATTACKS

Using the Packet Monitor statistics, the distribution of legitimate HTTP traffic and filtered traffic can be determined. In this case, the filter was set to catch the SynFlood attacks.

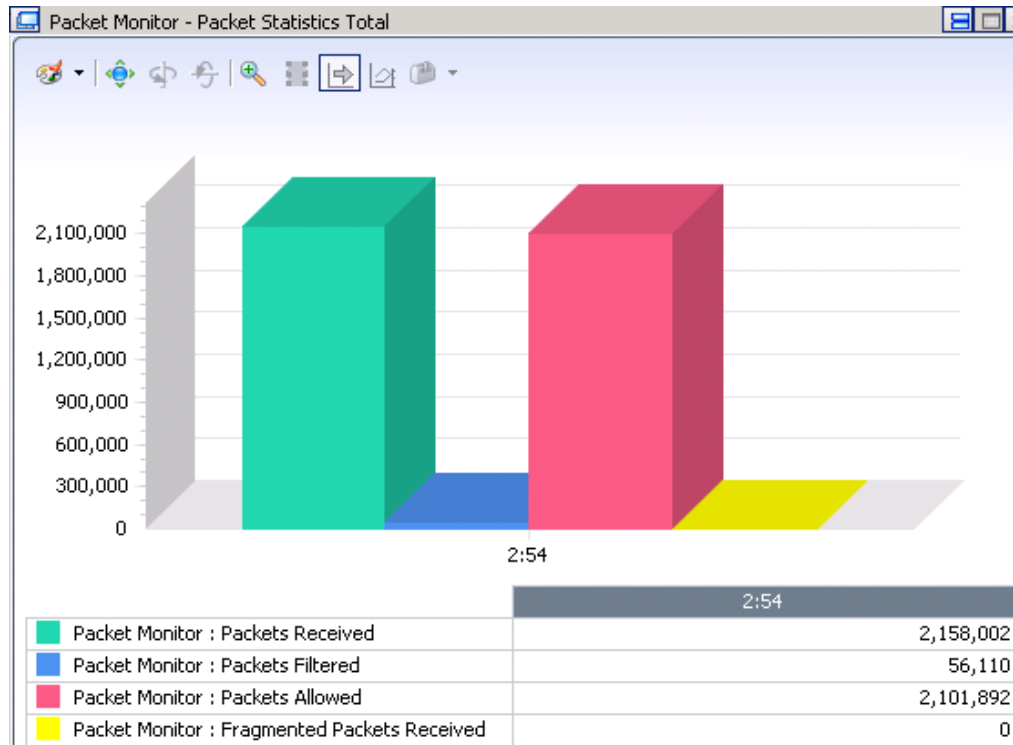


Figure 37. Packet Monitor Statistics View

Troubleshooting and Diagnostics

Issue	Diagnosis, Suggestions
A large number of TCP resets are received on client and server side throughout the test.	If there are continuous failures observed during steady-state operation, it is possible that the device is reaching saturation because of DoS attacks. This is very common during SYN flood attacks.
The throughput goes up and down.	If the device does not reach steady-state, check the TCP failures. High TCP timeout and RST packets can indicate that the device is unable to handle the load because of the DoS attacks.
The Simulated User count is increasing. The test tool is unable to reach target Throughput.	If the Simulated User count is increasing and the throughput is not met, it indicates that the test tool is actively seeking to reach the target. Check for TCP failures to indicate the effects of the DoS attack.

Test Case: Mitigation of TCP SYN DDoS attack

Overview

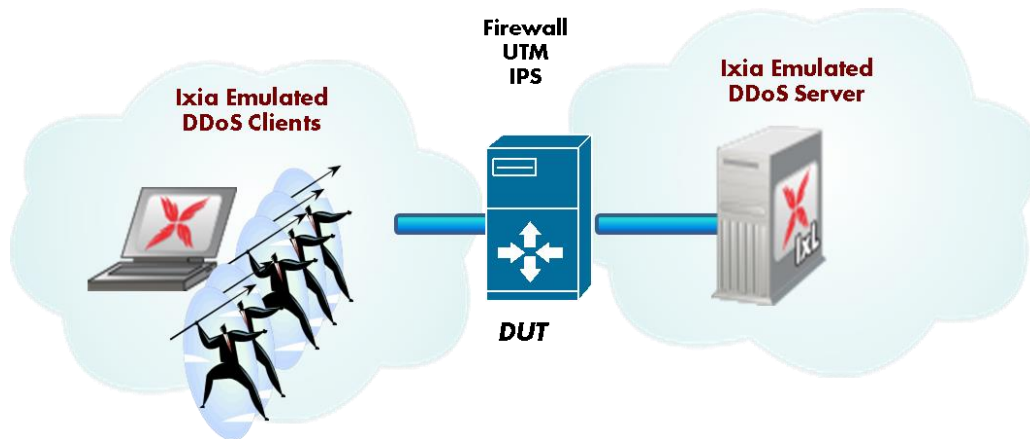
Network based DDoS attacks is one of the oldest methods of attacks yet still very effective and easy to implement. This test methodology walks you through a configuration that uses the TCP SYN Flooding attack to measure the mitigation capabilities of the intermediate firewalls. The example covered in this test case is performed using attacks injected at line rate on 1GE, 10GE or 40GE interfaces. It requires a pair of Ixia test ports.

Objective

The goal of this test is to measure DUT's capabilities to detect and mitigate the TCP SYN Flooding attack. You can also add the application traffic in addition to the DDoS traffic to assess the impact in quality of experience of the users using web, voice or video services.

Setup

The current setup consists of two Xcellon-Ultra NP ports connected directly to the tested device/system under test.



In this test topology, Ixia emulates:

- a BOTNET consisting of 100 DDoS Clients on **port1**
- a target network by placing a DDoS Server component on **port2**

This topology can be used to test intermediate devices such as firewalls and unified thread management systems. The DDoS Server activity is optional, and it can be replaced with an external target such as an Apache Web Server.

Including the Ixia DDoS Server has the following advantages:

TEST CASE: MITIGATION OF TCP SYN DDOS ATTACK

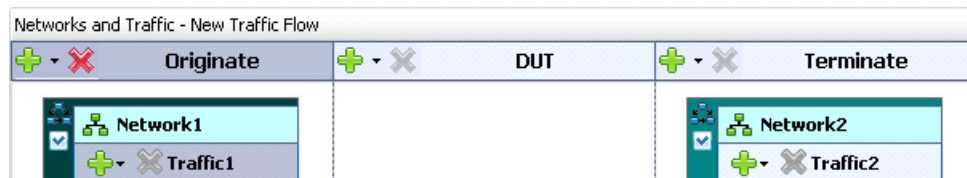
- It discards all DDoS traffic at FPGA level avoiding any impact on the CPU used by the target port where application servers may be emulated.
- It provides measurements such as successful attack frames, successful attack rate and attack throughput for analysis.

Step-by-Step Instructions

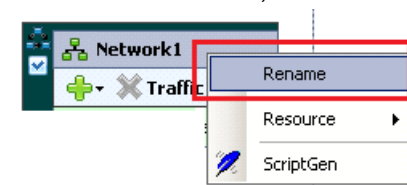
This section provides step by step instructions to execute this test.

1. Create and configure the **BOTNET (DDoS Clients) network and TARGET (DDoS Server) network**

- 1.1. Start IxLoad user interface.
- 1.2. Select **File | New ...** to create a new configuration.
- 1.3. Create two networks **Network1** and **Network2**.

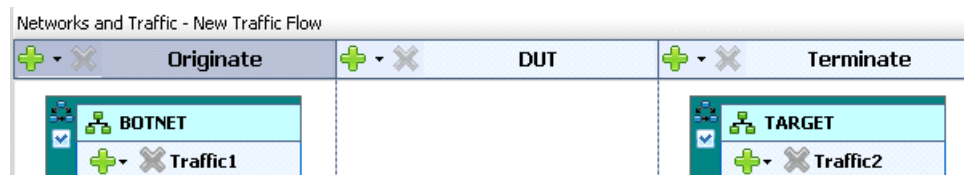


- 1.4. Rename **Network1** as **BOTNET** as follows.
 - a. Right-click the traffic object.
 - b. Select **Rename**; then enter the desired name, 'BOTNET'.



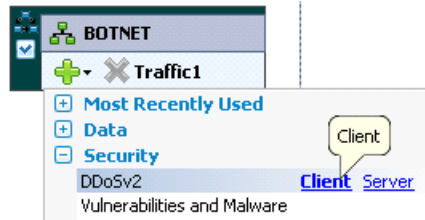
This network emulates multiple IPs flooding the **TARGET** network.

- 1.5. Rename **Network2** as **TARGET**. This network hosts the IP address(es) of the targeted victim(s).

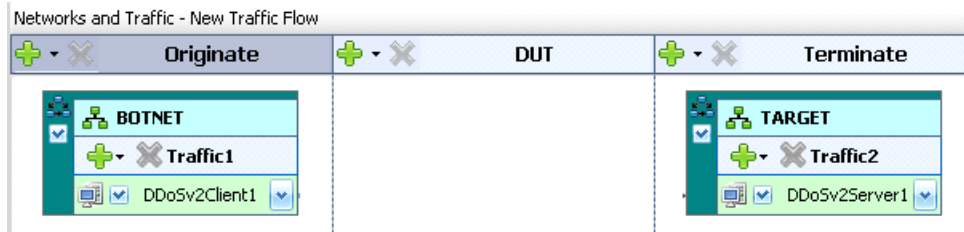


- 1.6. Add a **DDoSV2 Client** activity to the **BOTNET** network.

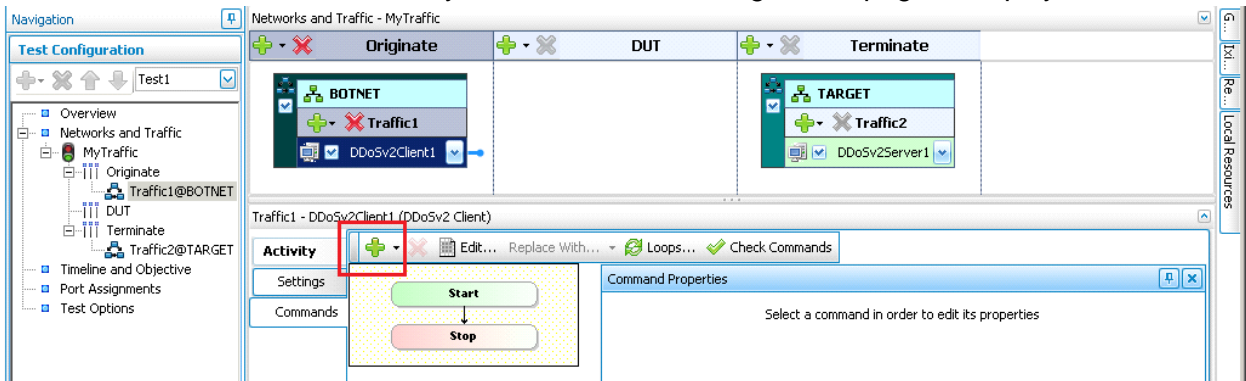
TEST CASE: MITIGATION OF TCP SYN DDOS ATTACK



1.7. Add a **DDoSV2 Server** activity to the **TARGET** network.




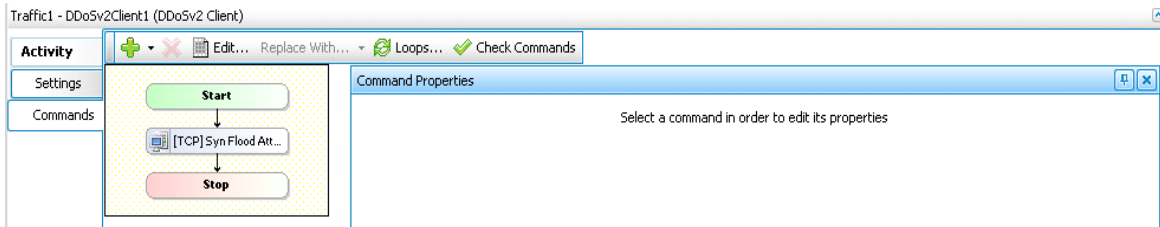
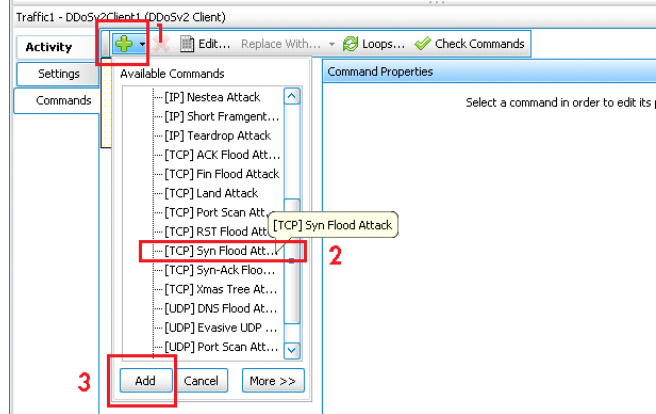
1.8. Select **DDoSV2Client1** activity; the DDoS client configuration page is displayed.



TEST CASE: MITIGATION OF TCP SYN DDOS ATTACK

1.9. Add the **[TCP] Syn Flood Attack** command as follows:

- a. Select  the **add command(s)** button (1).
- b. Select the **[TCP] Syn Flood Attack** command (2).
- c. Click **Add**. (3). The **[TCP] Syn Flood Attack** command is included to the command list.



1.10. Select the **[TCP] Syn Flood Attack** command. The Command properties for TCP Syn Flood Attack pane is displayed. Configure the following:

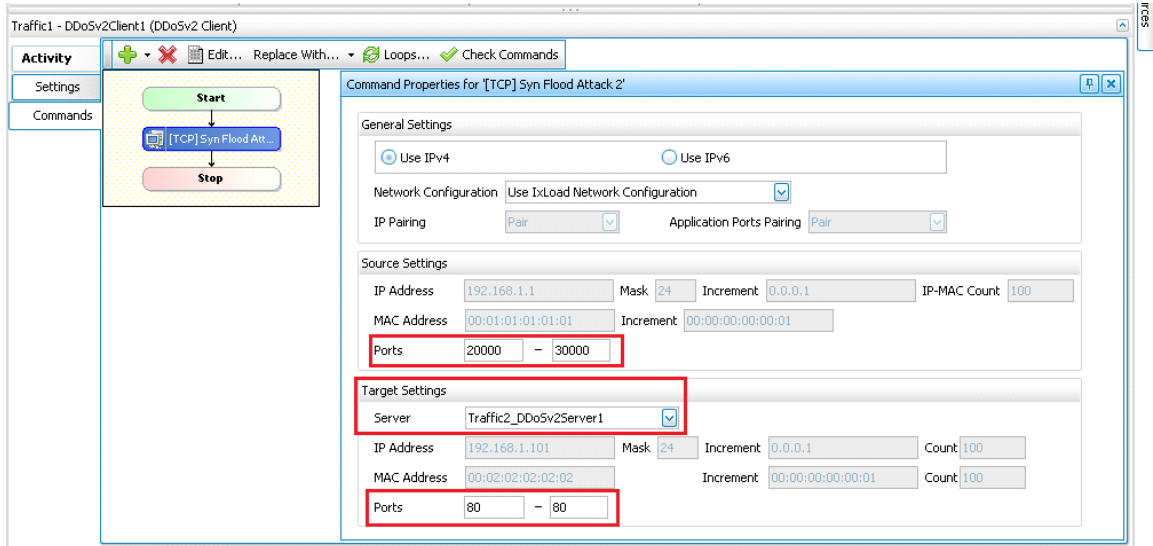
- a. Ensure that **Use IPv4** option is selected.
- b. Ensure that **Network Configuration** is set to **Use IxLoad Network Configuration**.

Note: You can use the **Custom Configuration** option to set spoofed IP addresses.

- c. Under **Source Settings** parameters group, set the source **Port(s)** as a range between **20,000** to **30,000**.

TEST CASE: MITIGATION OF TCP SYN DDOS ATTACK

- d. Under **Target Settings** parameters group, set:
 - i. the Server to **Traffic2_DDoSv2Server1**
 - ii. the destination **Port(s)** as a range between **80 to 80**



Note:

This configuration generates **TCP SYN** packets using all IP addresses added at the network level that are mapped to the **DDoSV2Client1** activity. The destination IP range is set to the range of IPs defined under **TARGET** network that are mapped to **DDoSV2Server1** activity. All packets are sent to port 80. For example, if **Source IPs = 10.10.10.1 + 100** and the **Destination IP(s) = 10.10.10.101**, then the packets are generated as follows:

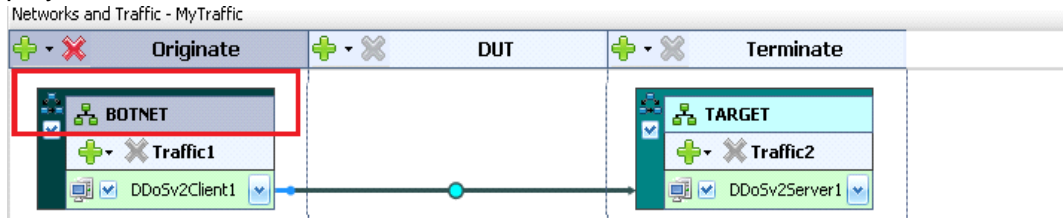
Packet #	Source IP	Source Port	Destination IP	Destination Port
1	10.10.10.1	20,000	10.10.10.101	80
2	10.10.10.1	20,001	10.10.10.101	80
3	10.10.10.1	20,002	10.10.10.101	80
...
10,000	10.10.10.1	30,000	10.10.10.101	80
10,001	10.10.10.2	20,000	10.10.10.101	80
10,002	10.10.10.2	20,001	10.10.10.101	80
...
20,000	10.10.10.2	30,000	10.10.10.101	80
...
99,000	10.10.10.99	30,000	...	80
99,001	10.10.10.100	20,000	10.10.10.101	80
99,002	10.10.10.100	20,001	10.10.10.101	80
...
100,000	10.10.10.100	30,000	10.10.10.101	80

TEST CASE: MITIGATION OF TCP SYN DDOS ATTACK

Packet #	Source IP	Source Port	Destination IP	Destination Port
100,001	10.10.10.1	20,000	10.10.10.101	80

2. Configure the IP parameters of the BotNet network and Target network

2.1. Select the BOTNET network, then select the IP stack; the IP configuration page is displayed.



2.2. Set the IP parameters as shown in the following table

Network Name	IP Type	Address	Mask	Count	Gateway
BOTNET (WAN)	IPv4	12.1.1.2	16	100	12.1.1.1

Enabled	Name	Status	IP Type	Address	Mask	Increment	Count	Gateway	Gateway Increment
<input checked="" type="checkbox"/>	IP-R1	Negotiated	IPv4	12.1.1.2	16	0.0.0.1	100	12.1.1.1	0.0.0.0

Figure 38.

Figure 39. BOTNET network - defining IP addresses

2.3. Select the TARGET network then select the IP stack; the IP configuration page is displayed.

TEST CASE: MITIGATION OF TCP SYN DDOS ATTACK

2.4. Set the IP parameters as shown in the following table

Network Name	IP Type	Address	Mask	Count	Gateway
TARGET	IPv4	13.1.1.2	16	1	13.1.1.1

Enabled	Name	Status	IP Type	Address	Mask	Increment	Count	Gateway	Gateway Increment
<input checked="" type="checkbox"/>	IP-R2	Negotiated	IPv4	13.1.1.2	16	0.0.0.1	1	13.1.1.1	0.0.0.0

Figure 40. TARGET network - defining IP addresses

3. Set up IxLoad to analyze the test results

You can determine the *attack rate* and *attack throughput* of the test using IxLoad as follows:

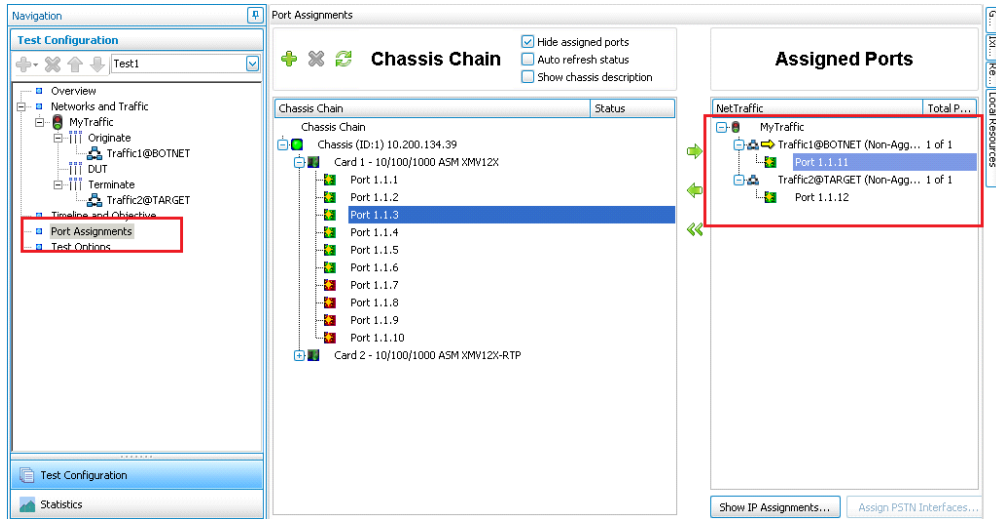
- 3.1. In the **Test Configuration** panel on the left, select **Timeline & Objective**.
- 3.2. In the **Timeline and Objective** pane on the right, set Objective Type to **Throughput (Gbps)**
- 3.3. Set **Objective Value** to 1.
- 3.4. In the **Timeline** pane, set **Sustain Time** to 5 minutes.

Network Traffic Mapping	Objective Type	Objective Value	Timeline	Iteration Time	Total Time
MyTraffic					
Traffic1@BOTNET	Throughput (Gbps)	1	Timeline1	000:05:40	000:05:40
DDoSV2Client1	Throughput (Gbps)	1	Timeline1	000:05:40	000:05:40
Traffic2@TARGET	N/A	N/A	<Match Longest>	000:05:40	000:05:40
DDoSV2Server1	N/A	N/A	<Match Longest>	000:05:40	000:05:40

3.5. Assign test ports to the BotNet network and Target network as follows:

- a. In the Test Configuration panel on the left, select **Port Assignments**.
- b. In the Port Assignments pane on the right, click **Add Chassis** button.
- c. Use the IP address of your chassis to add it to your configuration.
- d. Assign a port to the **BOTNET** network.
- e. Assign a port to the **TARGET** network.

TEST CASE: MITIGATION OF TCP SYN DDOS ATTACK



3.6. Define the other test options as follows:

- In the Test Configuration panel on the left, select Test Options.
- In the Test Options pane on the right, select:
 - Forcefully Take Ownership**
 - Reboot Ports before Configuring**
 - Release Configuration After Test**
- Set **CSV Polling Interval** to **2** seconds.
- Set **Throughput Stat Units** to **Mbps**
- Select **Enable Network Diagnostics** and select all group options.

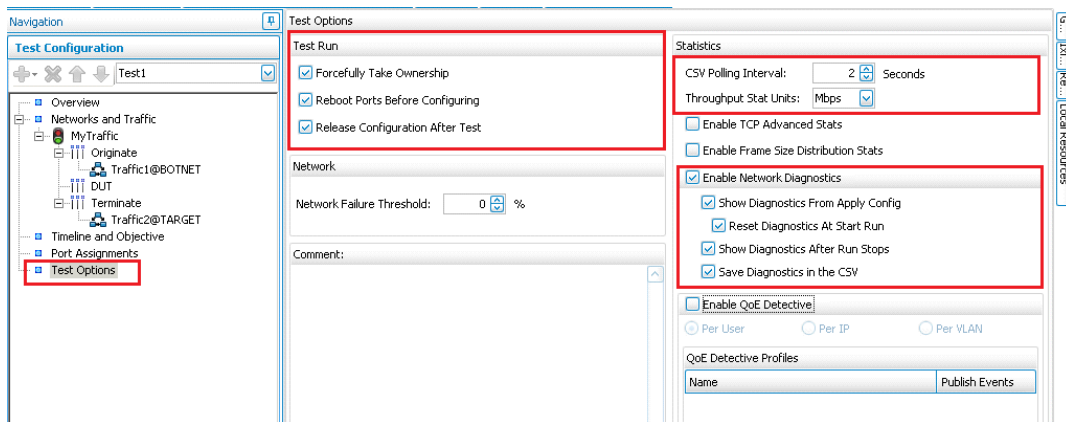


Figure 41. Test Options

3.7. Save your configuration file using **File | Save** or **File | Save As ...**

Example: **C:\Vxia\DDoS\DDoS-Pattern-001-TCP-SYN-Flooding.rxf**

TEST CASE: MITIGATION OF TCP SYN DDOS ATTACK

4. Run the test

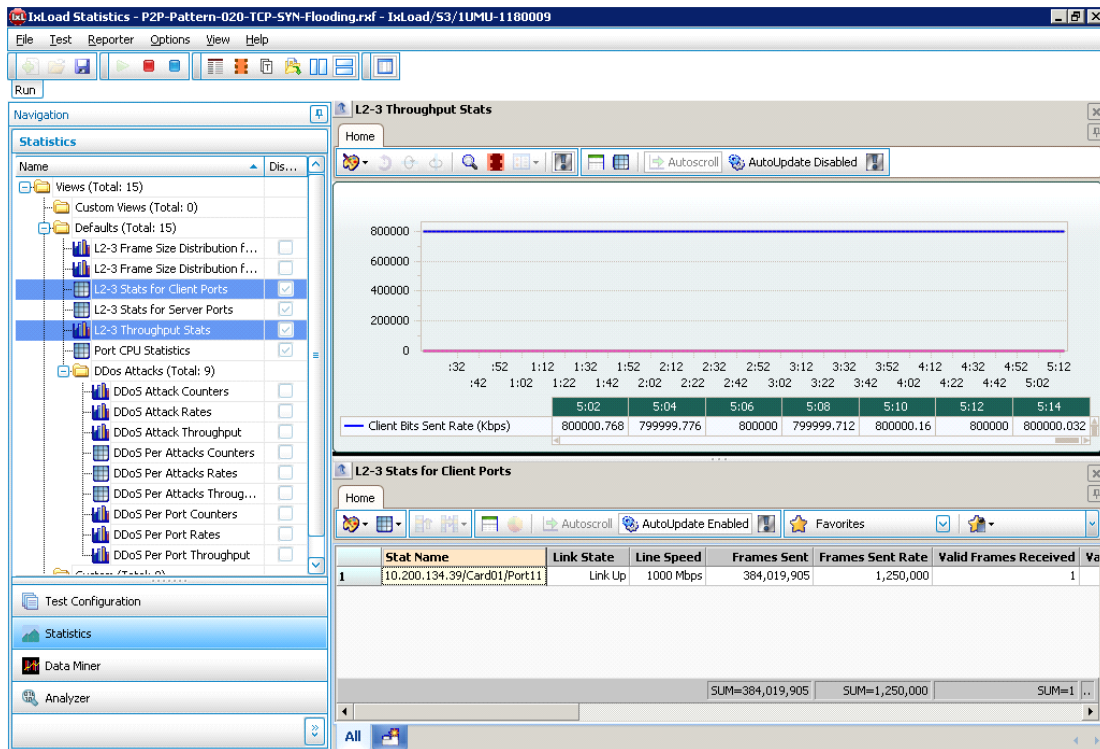
- 4.1. Run the test by selecting, **Test** menu | **Start** option, or by using the **START** button from toolbar.

Results analysis

The IxLoad application allows you to collect various statistics to suit your test requirements. Select the following key statistics to analyze the results for this test.

Network Statistics

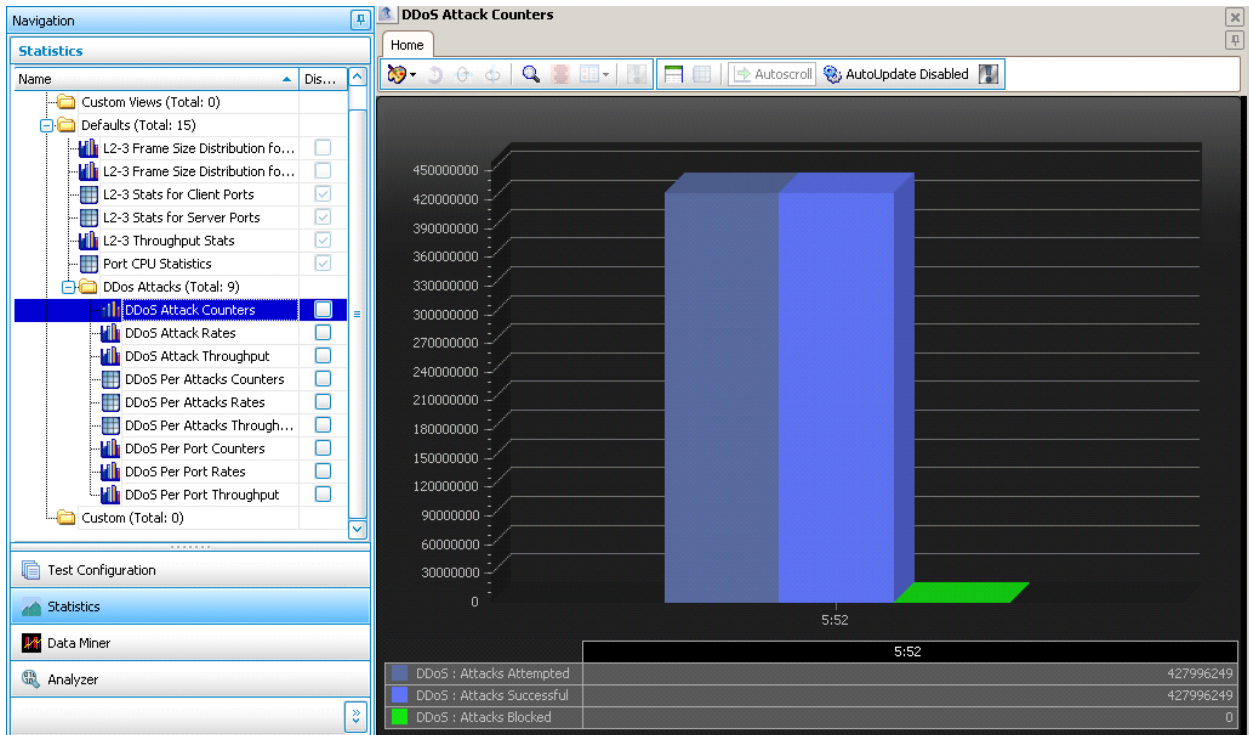
- From the IxLoad Statistics window, Statistics panel on the left, select **L2-3 Throughput** statistics view to monitor the L2 throughput.
- Select **L2-3 Stats for Client ports** to monitor the frames sent rate, frames sent, link speed, bytes sent, and bits sent rate, for the BOTNET network (DDoS Client).
- Select **L2-3 Stats for Server ports** to monitor the frames received rate, frames received, link speed, bytes received, bits received rate for the TARGET network (DDoS Server).



TEST CASE: MITIGATION OF TCP SYN DDOS ATTACK

DDoS Attack Statistics

From the IxLoad Statistics window, Statistics panel on the left, select the DDoS Attacks statistics. They include counters and rate statistics for attempted, successful and blocked attacks. By default, each metric is aggregated at test level, per attack and per port (test interface).



TEST CASE: MITIGATION OF TCP SYN DDOS ATTACK

DDoS Per Attacks Rates

Stat Name	5:32	5:34
1 SynFlood : Attacks Attempted/s	1,250,000.000	1,250,000.000
2 SynFlood : Attacks Successful/s	1,250,000.000	1,250,000.000
3 SynFlood : Attacks Blocked/s	0	0

DDoS Per Attacks Throughput

Stat Name	5:42
1 SynFlood : DDoS Throughput (Attempted kbps)	800,000.000
2 SynFlood : DDoS Throughput (Successful kbps)	800,000.000

DDoS Per Attacks Counters

Stat Name	5:38	5:38	5:40	5:42	5:44	5:46	5:48	5:50
1 SynFlood : Attacks Attempted	415,387,340	417,887,340	420,387,340	422,887,340	425,387,340	427,887,340	429,073,679	429,073,679
2 SynFlood : Attacks Successful	415,387,337	417,887,337	420,387,337	422,887,337	425,387,337	427,887,337	429,073,679	429,073,679
3 SynFlood : Attacks Blocked	0	0	0	0	0	0	0	0

DDoS Per Port Rates

Stat Name	5:38
1 10.200.134.39/Card1/Port11 : Attacks Attempted/s	1,250,000.000
2 10.200.134.39/Card1/Port11 : Attacks Successful/s	1,250,000.000
3 10.200.134.39/Card1/Port11 : Attacks Blocked/s	0

DDoS Per Port Counters

Stat Name	5:50
1 10.200.134.39/Card1/Port11 : Attacks Attempted	429,073,679
2 10.200.134.39/Card1/Port11 : Attacks Successful	429,073,679
3 10.200.134.39/Card1/Port11 : Attacks Blocked	0

DDoS Per Port Throughput

Stat Name	4:10	4:12	4:14	4:16	4:18	4:20	4:22	4:24
1 10.200.134.39/Card1/Port11 : DDoS Throughput (Attempted kbps)	800,000.000	800,000.000	800,000.000	800,000.000	800,000.000	800,000.000	800,000.000	800,000.000
2 10.200.134.39/Card1/Port11 : DDoS Throughput (Successful kbps)	800,000.000	800,000.000	800,000.000	800,000.000	800,000.000	800,000.000	800,000.000	800,000.000

TEST CASE: MITIGATION OF TCP SYN DDOS ATTACK

Test Variables

The IxLoad application offers the following test configuration parameters that provide you the flexibility to simulate the traffic profile that a device would experience in a production network.

Test tool variables

Parameter	Current Value	Alternative Settings
IP version	IPv4	IPv6
Botnet Size	100 unique IPs	Up to 128,000 IPs per port
Target Network	1 IP address (1 host count)	Range of IP addresses Increase the host count value on Target Network and adjust subnet mask and IP increment to match desired range
Source Port(s)	20000-30000	Decrease or Increase port range; combined with smaller/larger number of IP addresses can trigger higher/lower bandwidth per user
Destination Port(s)	80	Match the TCP port used by TCP applications used by the TARGET host (example: 21, 8080, 443, 22, so on). Use range of destination ports such as 20-21 to attack FTP services on TARGET
DDoS Pattern	TCP SYN Flooding	Use alternative TCP Flooding variants such as -- TCP SYN-ACK Flooding -- TCP FIN Flooding -- TCP Xmas Tree -- TCP Port Scan -- TCP Land Attack -- TCP ACK Flood -- TCP RST Flooding

Conclusions

This test case demonstrates how to configure the IxLoad application to determine the maximum attack rate and attack throughput that a DDoS mitigation system such as a firewall or an UTM can mitigate, while the system under test is being flooded with *TCP SYN Flooding*.

Test Case: Mitigation of ICMP Fragments DDoS Flooding attack

Overview

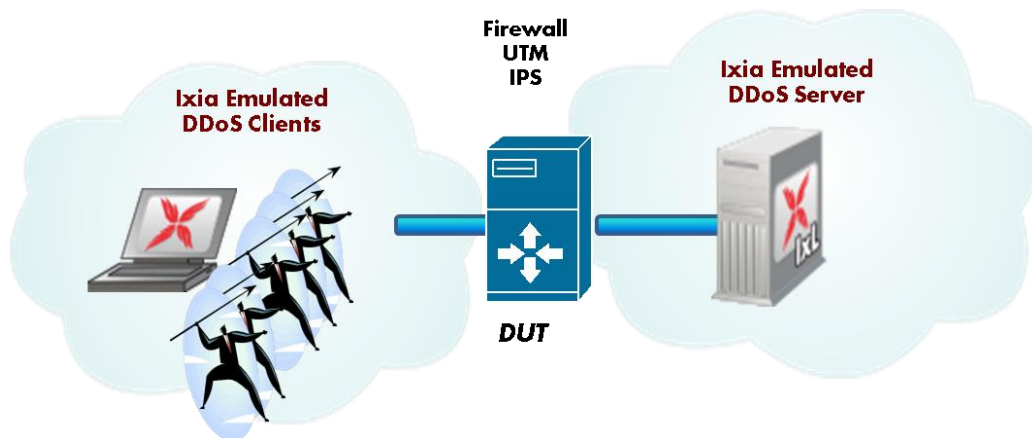
This test methodology walks you through a configuration that uses ICMP Fragments Flooding attack to measure the mitigation capabilities of intermediate firewalls. The example covered in this test case is performed using attacks injected at line rate on 1GE, 10GE or 40GE interface. It requires a pair of Ixia test ports.

Objective

The goal of this test case is to measure DUT's capabilities to detect and mitigate ICMP Fragments Flooding DDoS attack. Application traffic can be added in addition to the DDoS traffic to assess the impact in quality of experience of users using web, voice or video services.

Setup

The current setup consists of two Xcellon-Ultra NP ports connected directly to the tested device/system under test.



In this test topology, Ixia emulates:

- a BOTNET consisting of 100 DDoS Clients on **port1**
- a target network by placing a DDoS Server component on **port2**

This topology can be used to test intermediate devices such as firewalls and unified thread management systems. This test case is a modification of the **Mitigation of TCP SYN Flooding DDoS attack** test case described above.

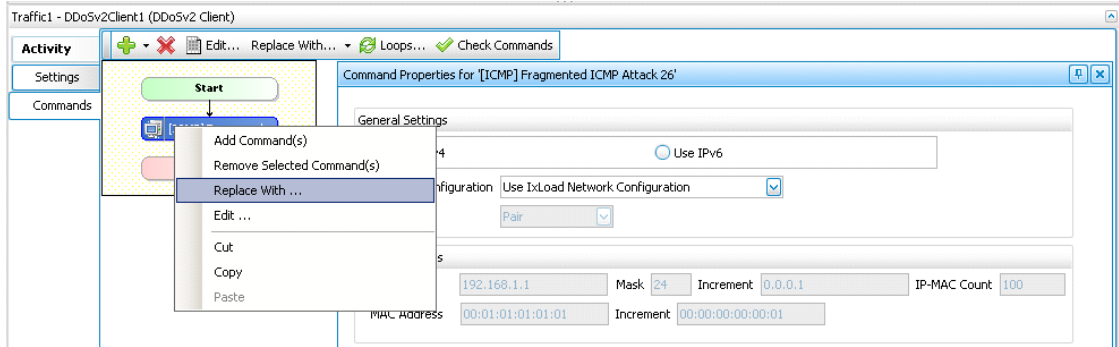
TEST CASE: MITIGATION OF A ICMP FRAGMENTS DDOS FLOODING ATTACK

Step-by-Step Instructions

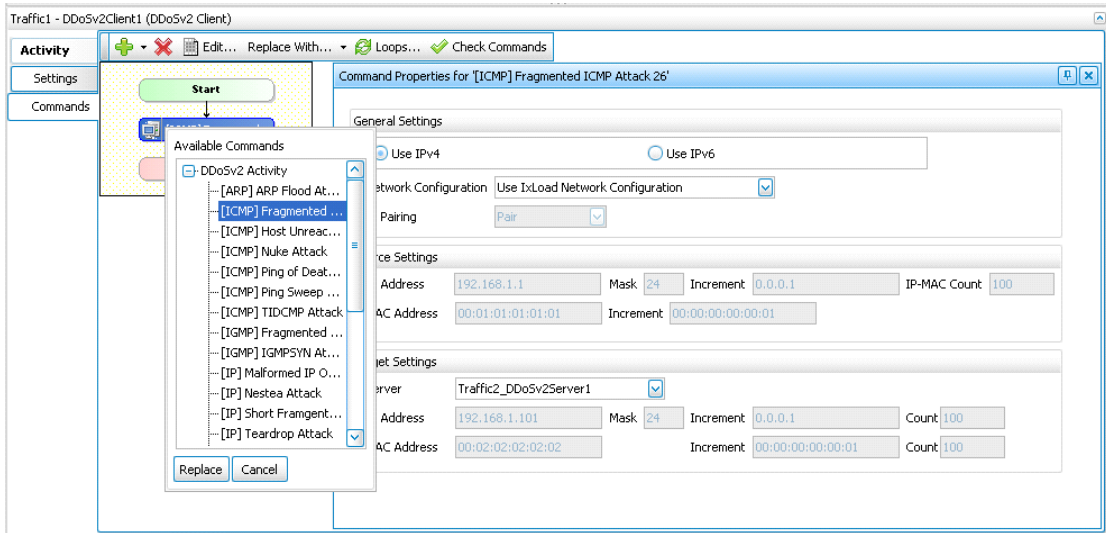
1. Modify the DDoS pattern from TCP SYN Flooding to Fragmented ICMP attack

1.1. Replace the [TCP] TCP SYN Flooding attack with [ICMP] Fragmented ICMP attack as follows:

- a. Right-click [TCP] TCP SYN Flooding command.
- b. Select **Replace With ...**.



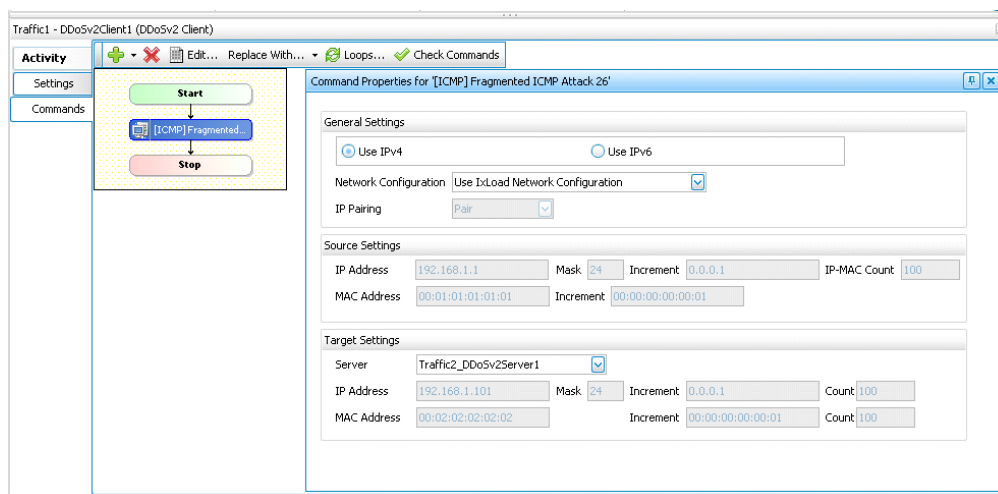
- c. Select **[ICMP] Fragmented ICMP Attack** from the DDoSv2 Activity drop-down list.



TEST CASE: MITIGATION OF A ICMP FRAGMENTS DDOS FLOODING ATTACK

1.2. Configure the **[ICMP] Fragmented ICMP attack** by setting the parameters shown below:

- In Command Properties for [ICMP] Fragmented ICMP Attack pane, ensure that **Use IPv4** is selected.
- Set the Network Configuration to **Use IxLoad Network Configuration**.
- Set the Target Server to use **Traffic2_DDoSv2Server1** as destination.



1.3. Save your configuration file using **File | Save As**.

Example: *C:\VXI\DDoS\Pattern-002-ICMP-Fragments-Flooding.rxf*

1.4. Run the test and compare the results with the previous ones.

TEST CASE: MITIGATION OF A ICMP FRAGMENTS DDOS FLOODING ATTACK

Test Variables

The IxLoad application offers the following test configuration parameters that provide you the flexibility to simulate a traffic profile that a device would experience in a production network.

Test tool variables

Parameter	Current Value	Alternative Settings
IP version	IPv4	IPv6
Botnet Size	100 unique IPs	Up to 128,000 IPs per port
Target Network	1 IP address (1 host count)	Range of IP addresses Increase the host count value on Target Network and adjust subnet mask and IP increment to match desired range
DDoS Pattern	ICMP Fragments Flooding	Use alternative IP Flooding variants such as 'ICMP Ping of Death', 'ICMP Host Unreachable', 'ICMP PING Sweep', or 'ICMP TIDCMP Attack' and 'ICMP Nuke Attack'

Conclusions

This test case demonstrates how to configure the IxLoad application to determine the maximum attack rate and attack throughput that a DDoS mitigation system such as a firewall or an UTM can mitigate, while the system under test is being flooded with *ICMP Fragments Flooding*.

Test Case: Mitigation of IP Short Fragments Flooding DDoS attack

Overview

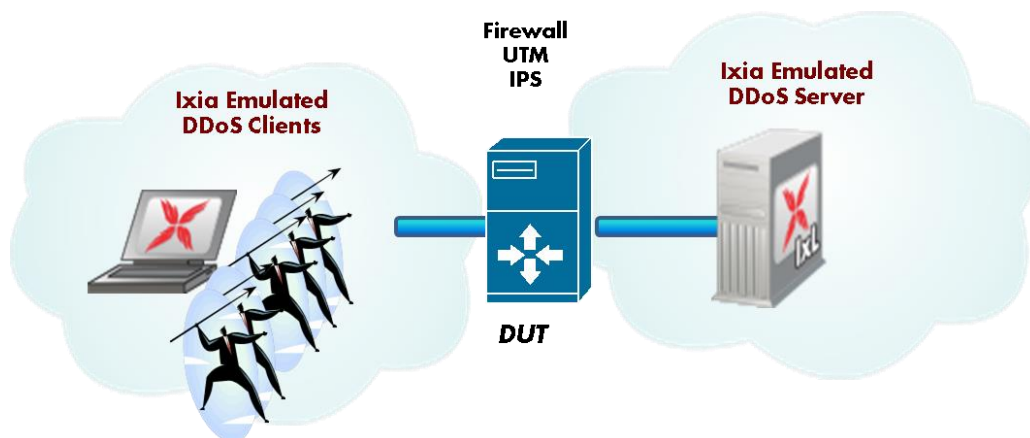
This test methodology walks you through a configuration that uses IP Short Fragments Flooding attack to measure the mitigation capabilities of intermediate firewalls. The example covered in this test case is performed using attacks injected at line rate on 1GE, 10GE or 40GE interfaces. It requires a pair of Ixia test ports.

Objective

The goal of this test case is to measure DUT's capabilities to detect and IP Short Fragments Flooding DDoS attack. Application traffic can be added in addition to the DDoS traffic to assess the impact in quality of experience of users using web, voice or video services.

Setup

The current setup consists of two Xcellon-Ultra NP ports connected directly to the tested device/system under test.



In this test topology, Ixia emulates:

- a BOTNET consisting in 100 DDoS Clients
- a target network by placing a DDoS Server

This topology can be used to test intermediate devices such as firewalls and unified thread management systems.

TEST CASE: MITIGATION OF IP SHORT FRAGMENT FLOODING DDOS ATTACK

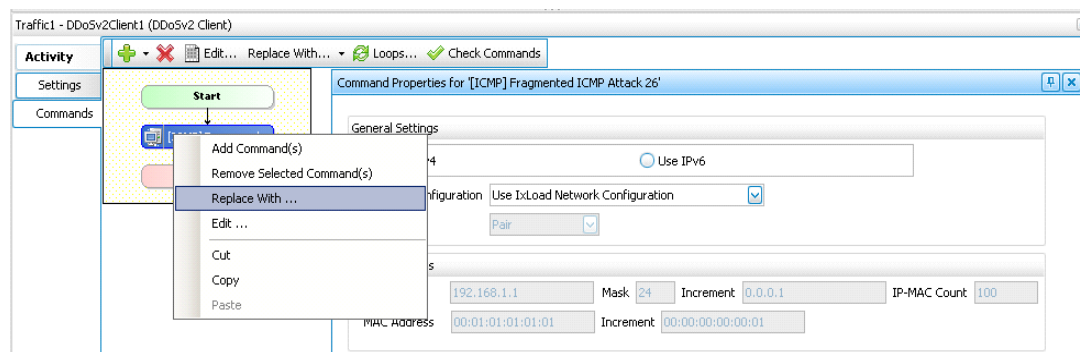
Step-by-Step Instructions

This test case is a modification of the Mitigation of TCP SYN Flooding DDoS attack test case described above.

1. Modify the DDoS attack pattern

1.1. Replace the **[TCP] TCP SYN Flooding attack** command with **[IP] Short Fragments attack** command as follows:

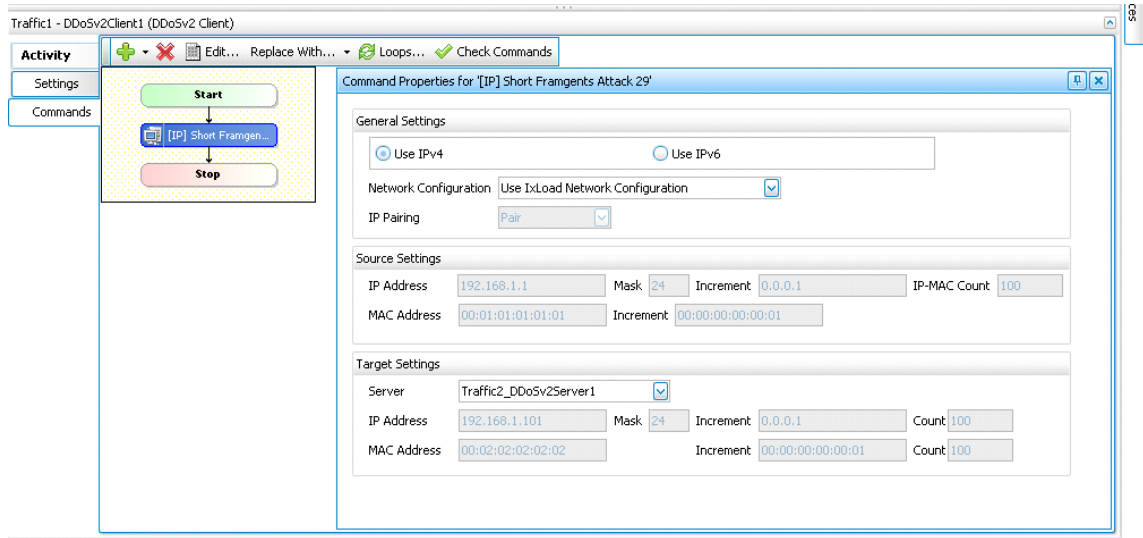
- a. Right click on **[TCP] TCP SYN Flooding** command.
- b. Select **Replace With ...** action.



- c. Select **[IP] Short Fragments** attack from the drop-down list.
- 1.2. Configure the **[IP] Short Fragments attack** by setting the parameters shown below:
- a. In Command Properties for [IP] Short Fragments Attack pane, ensure that **Use IPv4** is selected.
 - b. Set the Network Configuration to **Use IxLoad Network Configuration**.

TEST CASE: MITIGATION OF IP SHORT FRAGMENT FLOODING DDOS ATTACK

c. Set the Target Server to use **Traffic2_DDoSv2Server1** as destination.



1.3. Save your configuration file using **File | Save As**.

Example: *C:\IXIA\DDoS\Pattern-003-IP-Short-Fragments.rxf*

1.4. Run the test and compare the results with the previous ones.

Test Variables

The IxLoad application offers the following test configuration parameters that provide you the flexibility to simulate a traffic profile that a device would experience in a production network.

Test tool variables

Parameter	Current Value	Alternative Settings
IP version	IPv4	IPv6
Botnet Size	100 unique IPs	Up to 128,000 IPs per port
Target Network	1 IP address (1 host count)	Range of IP addresses Increase the host count value on Target Network and adjust subnet mask and IP increment to match desired range
DDoS Pattern	IP Short Fragments	Use alternative IP Flooding variants such as 'Malformed IP Options', 'Nestea' or 'IP Teardrop attack'

Conclusions

This test case demonstrates how to configure the IxLoad application to determine the maximum attack rate and attack throughput that a DDoS mitigation system such as a firewall or an UTM can mitigate, while the system under test is being flooded with *IP Short Fragments*.

TEST CASE: MITIGATION OF A DDoS MIX PATTERN USING EVEN TEST OBJECTIVE DISTRIBUTION OVER SAME TEST INTERFACE

Test Case: Mitigation of a DDoS MIX Pattern Using Even Test Objective Distribution Over Same Test Interface

Overview

This test methodology walks you through a configuration that uses the 3 DDoS attack patterns described in the previous test cases, namely:

- TCP SYN Flooding attack
- ICMP Fragments attack
- IP Short Fragments attack

In this test case, we combine the 3 attack patterns under same test activity and assess the distribution rate and throughput for each individual attack that is a part of the combination. When multiple attack patterns are added to an activity, the same test objective is evenly divided across each attack. The attack commands run in parallel, sharing the same objective timeline.

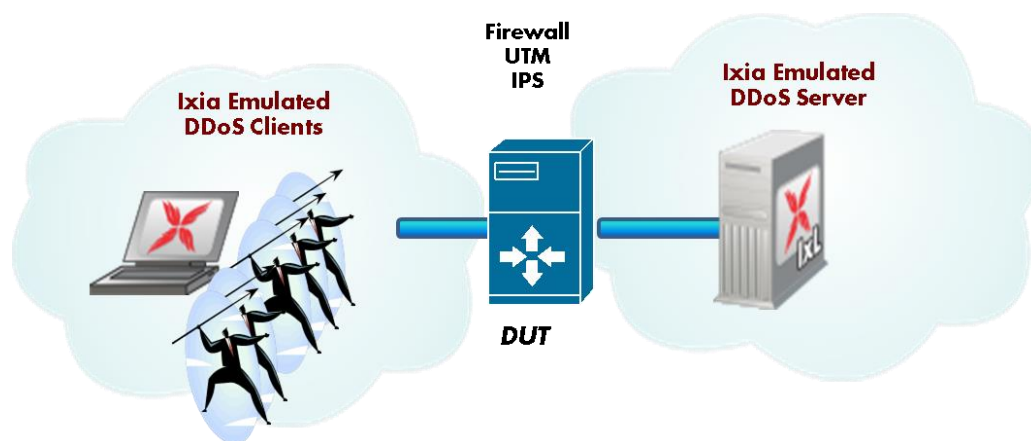
The example covered in this test case is performed using attacks injected at line rate on 1GE, 10GE or 40GE interfaces. It requires a pair of Ixia test ports.

Objective

The goal of this test case is to measure DUT's capability to detect and mitigate multiple attack patterns. Application traffic can be added in addition to the DDoS traffic to assess the impact in quality of experience of users using web, voice or video services.

Setup

The current setup consists of two Xcellon-Ultra NP ports connected directly to the tested device/system under test.



In this test topology, Ixia emulates:

TEST CASE: MITIGATION OF A DDOS MIX PATTERN USING EVEN TEST OBJECTIVE DISTRIBUTION OVER SAME TEST INTERFACE

- a BOTNET consisting in 100 DDoS Clients
- a target network by placing a DDoS Server

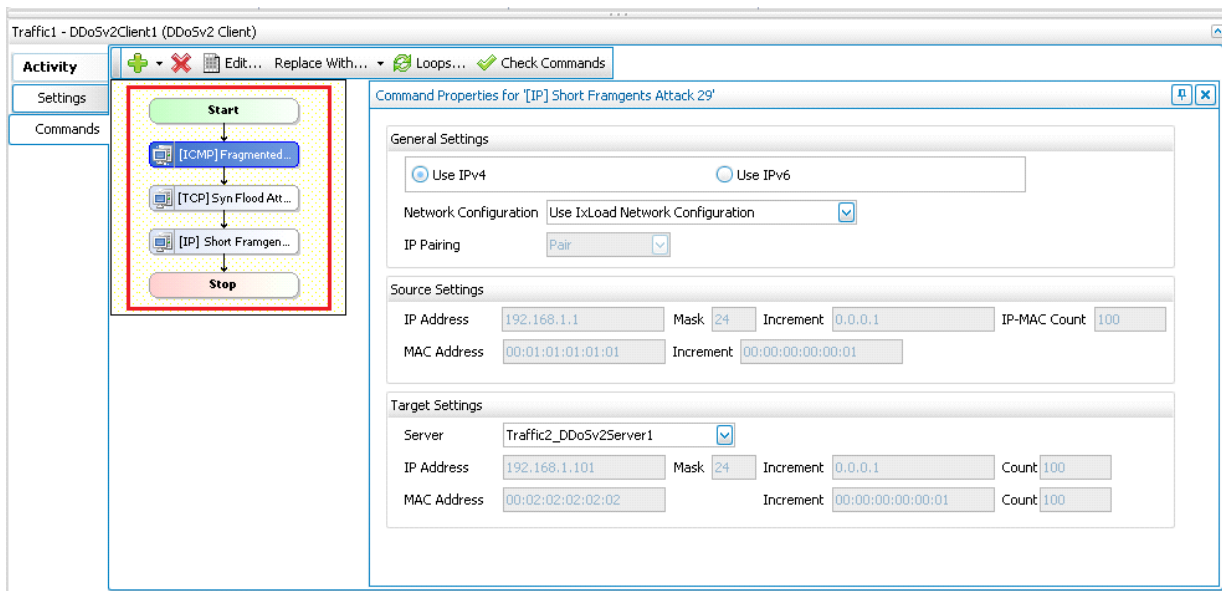
This topology can be used to test intermediate devices such as firewalls and unified thread management systems.

Step-by-Step Instructions

This test case is a modification of the **Mitigation of TCP SYN Flooding DDoS attack** test case.

To create this configuration, we can start with either one of the previous configuration (for example: **C:\VXIADDoS\Pattern-002-ICMP-Fragments-Flooding.rxf**)

1. Open the configuration **C:\VXIADDoS\Pattern-002-ICMP-Fragments-Flooding.rxf** using **File menu| Open ...** action. This configuration already includes the **[ICMP] Fragmented ICMP** attack.
Skip this step if your previous configuration is still open.
2. Add the second command, **TCP SYN Flooding** attack.
3. Add the third command, **IP Short Fragments** attack.



TEST CASE: MITIGATION OF A DDOS MIX PATTERN USING EVEN TEST OBJECTIVE DISTRIBUTION OVER SAME TEST INTERFACE

4. Use the following table to configure the settings for each attack command, refer to the configuration snapshots given below.

	[ICMP] Fragmented ICMP	[TCP] SYN Flooding	[IP] Short Fragments
IP version	IPv4		
Source IP(s)	Use IxLoad Network Configuration		
Source Port(s)	n/a	20,000-30,000	n/a
Target Server	Traffic2_DDoSv2Server1		
Target Port(s)	n/a	80	n/a

[ICMP] Fragmented ICMP Attack

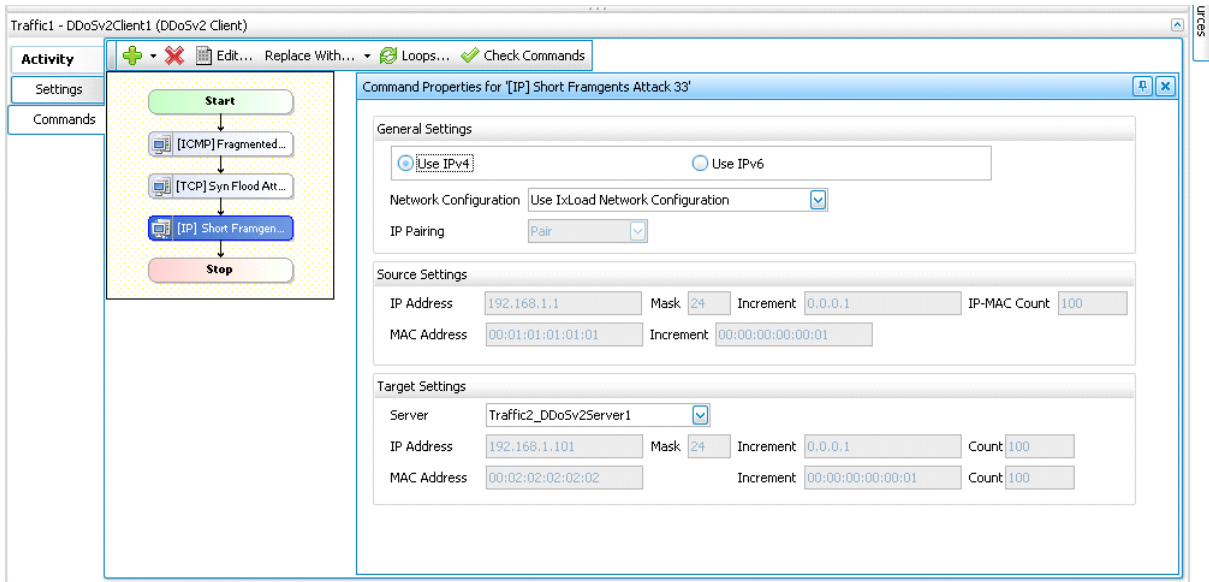
The screenshot shows the configuration for a [ICMP] Fragmented ICMP Attack. The 'Command Properties' window is set to Use IPv4, Use IxLoad Network Configuration, and Pair IP Pairing. Source settings include IP Address 192.168.1.1, Mask 24, Increment 0.0.0.1, and IP-MAC Count 100. Target settings include Server Traffic2_DDoSv2Server1, IP Address 192.168.1.101, Mask 24, Increment 0.0.0.1, and Count 100.

[TCP] SYN Flood Attack

The screenshot shows the configuration for a [TCP] SYN Flood Attack. The 'Command Properties' window is set to Use IPv4, Use IxLoad Network Configuration, and Pair IP Pairing. Source settings include IP Address 192.168.1.1, Mask 24, Increment 0.0.0.1, and IP-MAC Count 100. Target settings include Server Traffic2_DDoSv2Server1, IP Address 192.168.1.101, Mask 24, Increment 0.0.0.1, and Count 100. The Ports field is set to 20000 - 30000.

TEST CASE: MITIGATION OF A DDOS MIX PATTERN USING EVEN TEST OBJECTIVE DISTRIBUTION OVER SAME TEST INTERFACE

[IP] Short Fragments Attack



5. Save your configuration file using **File | Save As ...** at the following location:
C:\VIXIA\DDoS\Pattern-DDoS-MIX-3-Attacks.rxf
6. Run the test and compare the results with the previous ones.
7. Compare the distribution of attack rates & attack throughput across the attack patterns.

Test Variables

The IxLoad application offers the following test configuration parameters that provide you the flexibility to simulate a traffic profile that a device would experience in a production network.

Test tool variables

Parameter	Current Value	Alternative Settings
IP version	IPv4	IPv6
Botnet Size	100 unique IPs	Up to 128,000 IPs per port
Target Network	1 IP address (1 host count)	Range of IP addresses Increase the host count value on Target Network and adjust subnet mask and IP increment to match desired range
DDoS Pattern	-- TCP SYN Flooding -- ICMP Fragments -- IP Short Fragments	Use alternative mix of DDoS patterns (including all attacks)

TEST CASE: MITIGATION OF A DDOS MIX PATTERN USING EVEN TEST OBJECTIVE DISTRIBUTION OVER SAME TEST INTERFACE

Conclusions

This test case demonstrates how to configure the IxLoad application to determine the maximum attack rate and attack throughput that a DDoS mitigation system such as a firewall or an UTM can mitigate (block, while the system under test is being exposed to a mix of attack patterns using an even test objective distribution.

TEST CASE: MITIGATION OF A DDOS MIX PATTERN USING UNEVEN TEST OBJECTIVE DISTRIBUTION OVER SAME TEST INTERFACE

Test Case: Mitigation of a DDoS MIX Pattern Using Uneven Test Objective Distribution over Same Test Interface

Overview

This test methodology walks you through a configuration that uses the 3 DDoS attack patterns described in the previous test case. It uses a configuration that allows the control of the test objective on a per DDoS attack pattern basis using different ratios between:

- TCP SYN Flooding attack
- ICMP Fragments attack
- IP Short Fragments attack

To gain control over the traffic transmitted using each pattern in the test case, we combine the 3 DDoS client activities under the same test activity. Each DDoS activity includes a single attack pattern. Each activity has its own test objective, allowing control over the volume or rate of attacks to be transmitted. When multiple attack patterns are added to an activity, the same test objective is evenly distributed across each attack.

The example covered in this test case is performed using attacks injected at line rate on 1GE, 10GE or 40GE interfaces. It requires a pair of Ixia test ports.

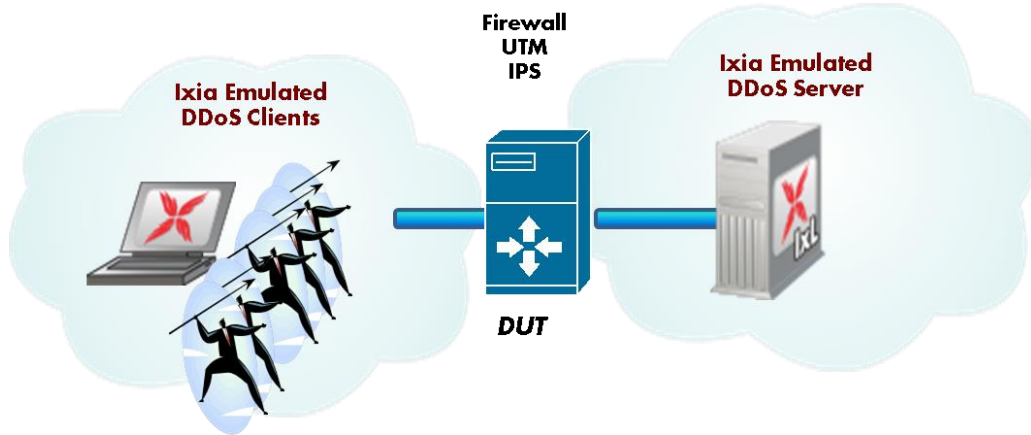
Objective

The goal of this test case is to measure DUT's capability to detect and mitigate a mix of multiple attack patterns with each pattern having its own test objective. Application traffic can be added in addition to the DDoS traffic to assess the impact in quality of experience of users using web, voice or video services.

TEST CASE: MITIGATION OF A DDoS MIX PATTERN USING UNEVEN TEST OBJECTIVE DISTRIBUTION OVER SAME TEST INTERFACE

Setup

The current setup consists of two Xcellon-Ultra NP ports connected directly to the tested device/system under test.



In this test topology Ixia emulates:

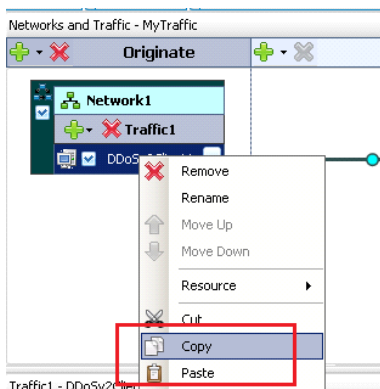
- a BOTNET consisting in 100 DDoS Clients
- a target network by placing a DDoS Server

This topology can be used to test intermediate devices such as firewalls and unified thread management systems.

Step-by-Step Instructions

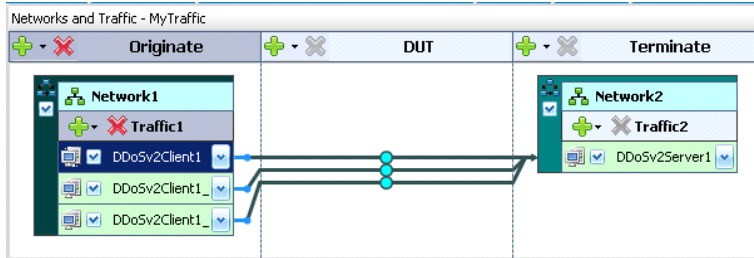
This test case is a modification of the *Mitigation of a DDoS MIX pattern using even test objective distribution over same test interface attack* test case.

1. Add DDoS client activities (one per DDoS pattern) as follows:
 - 1.1. Open the RXF from the previous test case.
 - 1.2. **Right click** on the **DDoSClient1** activity to display the menu, and then select the **Copy** command.

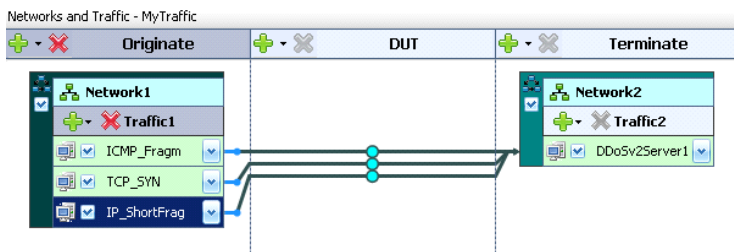


TEST CASE: MITIGATION OF A DDOS MIX PATTERN USING UNEVEN TEST OBJECTIVE DISTRIBUTION OVER SAME TEST INTERFACE

- 1.3. Right-click again on DDoSClient1 activity to display the menu, then select the **Paste** command to duplicate the activity. A copy of DDoSClient1 activity is created.
- 1.4. Repeat #1.3 to create a secondary copy. The result is shown below. Now each activity has the 3 attack patterns configured.



2. Rename first activity as **ICMP_Frag**.
3. Rename second activity as **TCP_SYN**.
4. Rename the third activity as **IP_ShortFrag**.



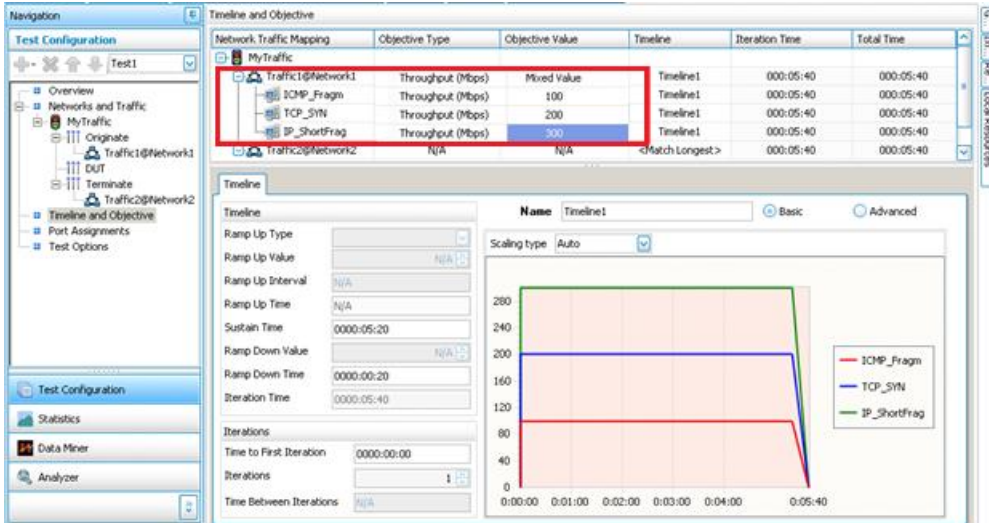
5. Select the first activity and delete the **[TCP] SYN Flooding** and **[IP] Short Fragments** attacks.
6. Select the second activity and delete the **[ICMP] Fragments ICMP** and **[IP] Short Fragments** attacks
7. Select the third activity and delete the **[ICMP] Fragments ICMP** and **[TCP] SYN Flooding** attacks

Notes:

- Each activity is now configured to generate a unique attack pattern
- Each activity has its own test objective, allowing granular control of the test objective value per attack.
- We use a single DDoS Server activity as a target.

TEST CASE: MITIGATION OF A DDoS MIX PATTERN USING UNEVEN TEST OBJECTIVE DISTRIBUTION OVER SAME TEST INTERFACE

8. Run the test using the following mix test objective:
 - a. ICMP Fragments = 100 Mbps
 - b. TCP SYN = 200 Mbps
 - c. IP_ShortFrag = 300 Mbps



9. Save your configuration file using **File | Save As ...** at the following location:
C:\VIXIA\DDoS\DDoS-Mix-PerAttackObjective.rxf
10. Compare the attack rates for each attack pattern.

Test Variables

The IxLoad application offers the following test configuration parameters that provide you the flexibility to simulate a traffic profile that a device would experience in a production network.

Test tool variables

Parameter	Current Value	Alternative Settings
IP version	IPv4	IPv6
Botnet Size	100 unique IPs	Up to 128,000 IPs per port
Target Network	1 IP address (1 host count)	Range of IP addresses Increase the host count value on Target Network and adjust subnet mask and IP increment to match desired range
DDoS Pattern	-- TCP SYN Flooding -- ICMP Fragments -- IP Short Fragments	Use alternative mix of DDoS patterns (including all attacks)
Application Traffic	None	HTTP, FTP, e-mail, voice & video

TEST CASE: MITIGATION OF A DDOS MIX PATTERN USING UNEVEN TEST OBJECTIVE DISTRIBUTION OVER SAME TEST INTERFACE

Conclusions

This test case demonstrates how to configure the IxLoad application to determine the maximum attack rate and attack throughput that a DDoS mitigation system such as a firewall or a UTM can mitigate (block), while the system under test is being exposed to a mix of DDoS patterns combined at different ratios.

Test Methodologies: IPsec VPN

IPsec Overview

The purpose of the IPsec and IKE protocols is to provide privacy, encryption, and data integrity for network traffic traveling over an insecure network, such as the Internet. Two forms of IPsec usage normally apply:

- Site to Site.** This is shown in Figure 42. Two sites are connected through a pair of IPsec secure gateways. The LANs at each location are presumed to be secure and the insecure segment between the secure gateways is secured through the use of the tunnel.

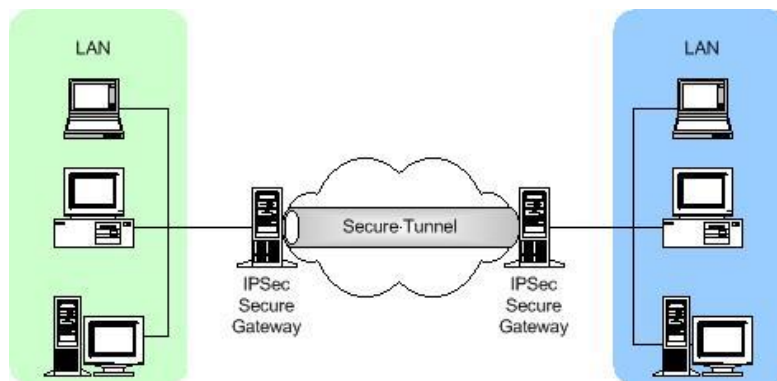


Figure 42. Site to site IPsec network

- Remote Access.** This is shown in Figure 43. In this case, the client is actually operating as its own secure gateway.

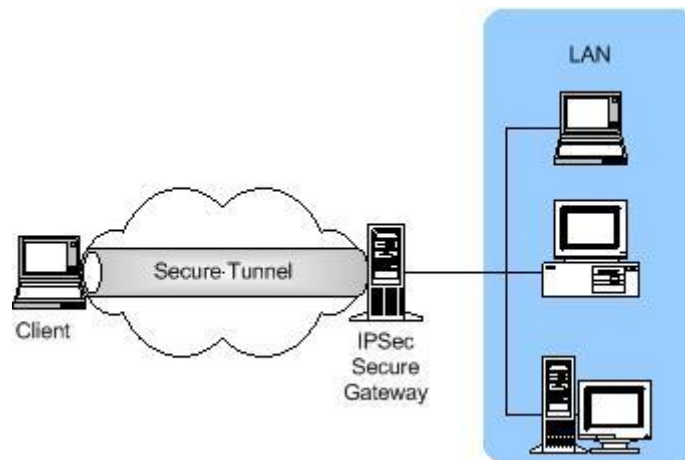


Figure 43. Remote access IPsec network

IPsec

IPsec ensures that network traffic is transmitted securely. It consists of a suite of protocols that ensure data integrity, data authenticity, data confidentiality, and data non-repudiation at the IP layer. Two IPsec protocols, AH and ESP, add headers (and in the case of ESP, a trailer) to each packet:

AH: authentication header. The AH protocol uses a hashing algorithm over a portion of the packet to ensure that the packet has not been modified during transit. AH provides data authenticity, data integrity, and data non-repudiation, but not data confidentiality.

ESP: encapsulated security payload. The ESP protocol introduces a portion of the original packet that has been encrypted, and adds a trailer to the end of the packet. ESP provides the same data protection as AH and in addition, it provides data confidentiality by encrypting the upper-layer payload.

The headers can be used separately or both can be used at the same time. The manner in which these headers are used is influenced by the two modes in which IPsec can operate:

Transport mode: In transport mode, an AH or ESP header is inserted between the IP header and the upper layer protocol header. See Figure 44.

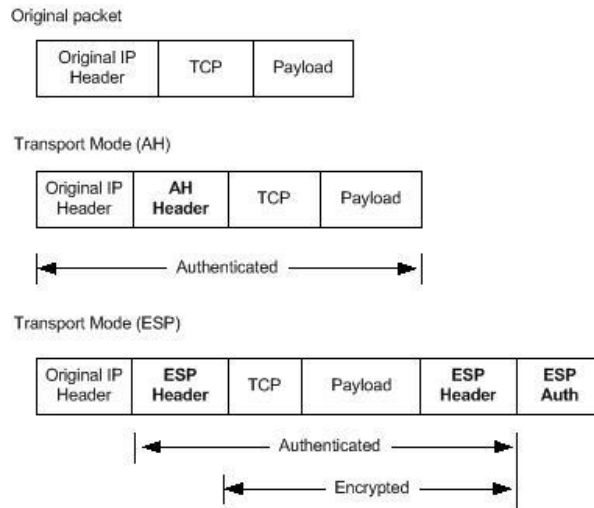


Figure 44. Transport mode packet format

The AH header includes a cryptographic checksum over the entire packet. The receiving end can verify that the entire packet was received without error or modification. The ESP header also includes a cryptographic checksum, and in addition, the packet's payload section is encrypted.

Transport mode is used only in remote access connections, where the source of the packets is also the crypto-endpoint or tunnel endpoint.

Tunnel mode: The original packet is encapsulated into a new packet that includes a new header and AH or ESP headers, as shown in Figure 45.

TEST METHODOLOGIES: IPSEC VPN

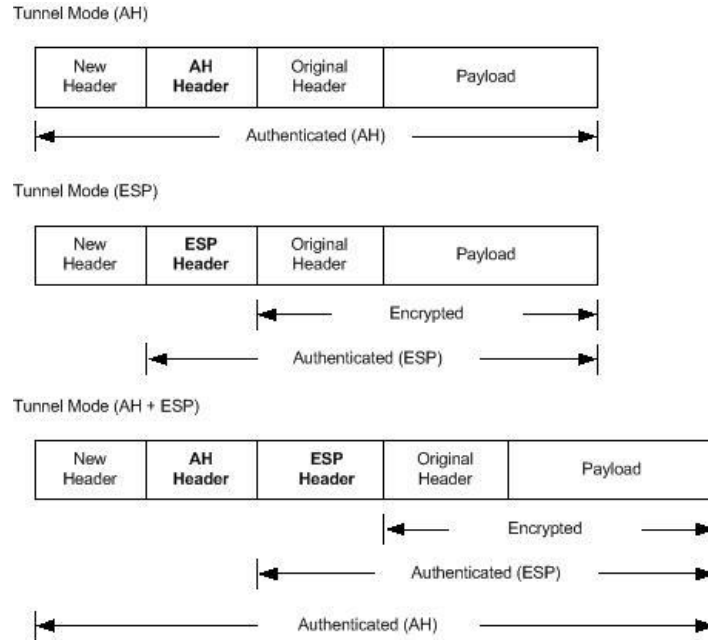


Figure 45. Tunnel mode packet format

The AH header is used to authenticate the entire packet. The ESP header is used to encrypt and authenticate the original packet. When used in combination, the original packet is encrypted and the entire packet is authenticated.

Tunnel mode is used when a security gateway is used to perform IPsec operation on behalf of a client computer, as is the case in LAN-to-LAN IPsec networks. The use of both AH and ESP headers provides maximum protection.

IKEv1

The purpose of the IKE protocol is to set up the parameters that allow two IPsec endpoints to communicate securely with each other. The set of parameters is called a security association (SA). SAs can be unidirectional or bidirectional.

The negotiation process between IPsec endpoints involves one party acting as an initiator and the other acting as a responder. Where parameters are being negotiated, the initiator offers the set of authentication, encryption, and other techniques that it is ready to use with the other endpoint. The responder tries to match this list against its own list of supported techniques. If there is any overlap, it responds with the common subset. The initiator chooses one combination of techniques from the responder and they proceed with the negotiated setting.

IKE negotiation is broken down into two phases:

Phase 1. Allows two gateways (one of which may be a client acting as its own gateway) to authenticate each other and establish communications parameters for phase 2 communications.

Phase 2. Allows two gateways to agree on IPsec communications parameters on behalf of sets of hosts on either side of the gateway.

Phase 1

During phase 1 negotiation, two endpoints authenticate each other. Based on policies enforced at each end, they decide that the other party is to be trusted. The means by which two parties trust each other can be based on one or more data items, including the following:

- Pre-shared key
- RSA-encrypted nonces
- Digital certificates using X.509

The endpoints also agree on the particular authentication and encryption algorithms to use when exchanging their later stage phase 1 and all phase 2 messages.

Two endpoints use a single bidirectional SA at the end of their phase 1 negotiation. There are two phase 1 negotiation modes:

Aggressive Mode. Three messages are exchanged to arrive at the bidirectional SA.

Main Mode. Six messages are exchanged to negotiate the SA. Main mode differs from aggressive mode in that the transmitted identities used for authentication are encrypted as part of the protocol. This keeps the identities of the two endpoints secret.

A single phase 1 SA may be used to establish any number of phase 2 SAs. During the negotiation process, the two endpoints generate a shared secret that is used to encrypt their communications. This shared secret is generated using public-private key cryptography in which two parties can generate a common data string without explicitly transmitting that data.

TEST METHODOLOGIES: IPSEC VPN

The set of parameters negotiated during phase1 are described in the following table.

Phase 1 negotiated parameters

Parameter	Usage
Mode	The basic mode of phase 1 communications: Aggressive mode: Three messages exchanged without identity protection. Main mode: Six messages exchanged with identity protection.
DH Group	The public-private cryptography used to create the shared secret uses an algorithm called Diffie-Hellman. DH Groups are different bit length selections used in this calculation.
Encryption Algorithm	The encryption algorithm used to protect communications during phase 1 and phase 2 message exchange. The two most common algorithms in use today are as follows: 3DES: A 168-bit algorithm using the Digital Encryption Standard (DES) three times. AES: The Advanced Encryption Standard, which may be used in any bit length. Common bit lengths are 128 and 256 bit.
Authentication Algorithm	The cryptographic checksum used over the packet to ensure data integrity. The two most common algorithms are as follows: MD5 SHA-1
SA Lifetime	The negotiated SA must be renegotiated after a period of time. This lifetime is itself negotiated.

Phase 2

In phase 2, each of the crypto endpoints attempts to negotiate the following SAs:

An ISAKMP SA: A bidirectional SA used to dynamically establish a secure channel for the negotiation of IPsec SAs.

An outbound IPsec SA: A unidirectional SA used to protect IPsec traffic sent to the remote tunnel endpoint.

An inbound IPsec SA: A unidirectional SA used to process IPsec traffic received from a remote crypto endpoint.

Phase 2 messages operate under the protection of a phase 1 SA by using the negotiated shared secret between the gateways. In addition to negotiating authentication and encryption

TEST METHODOLOGIES: IPSEC VPN

parameters, they also contribute random data to be used in generating the keys for the encryption algorithms that encrypt the payload data.

The following table describes the parameters for phase 2 negotiation.

Phase 2 negotiated parameters

Parameter	Usage
Mode	The basic mode of phase 2 communications: Transport mode: The original packet header is preserved. Used only for Client to LAN IPsec networks. Tunnel mode: Security gateways encapsulate and encrypt the original packet.
DH Group	This is needed only if a feature called Perfect Forward Security (PFS) is enabled. PFS generates a new shared secret with each phase 2 negotiation. The public-private cryptography used to create the shared secret using an algorithm called Diffie-Hellman. DH Groups are different bit length selections used in this calculation.
Encryption Algorithm	The encryption algorithm used to encrypt the data stream between the gateways during IPsec communications. The two most common algorithms in use today are as follows: 3DES: A 168-bit algorithm using the Digital Encryption Standard (DES) three times. AES: The Advanced Encryption Standard, which may be used in any bit length. Common bit lengths are 128 and 256 bit.
Authentication Algorithm	The cryptographic checksum used over the packet to ensure data integrity. The two most common algorithms are as follows: MD5 SHA-1
SA Lifetime	The negotiated SA must be renegotiated after a period of time. This lifetime is itself negotiated.

Xauth and Modcfg

IKE Extended Authentication (Xauth) is an enhancement to the existing IKE protocol. Xauth is not a replacement for IKE; it is an extension of it. While IKE performs device authentication, Xauth performs user authentication. Xauth user authentication occurs after IKE authentication

phase 1, but before IKE IPsec SA negotiation phase 2. With Xauth, after a device has been authenticated during normal IKE authentication, IKE can then authenticate a user using that device.

Modecfg (mode-configuration) is an IPsec feature that functions like DHCP for IPsec clients. Modecfg enables a responder to send (push) addresses (such as a private IP address, a DNS server's IP address) to an initiator.

Modecfg can also work in the opposite direction, with the client retrieving (pull) address information from the server. Modecfg is typically used in remote-access scenarios, where addresses may be part of a pool, with different privileges given to different addresses, or groups of addresses. The responder (the device supplying addresses) sends the addresses during the IKE key exchange.

IPComp

IP compression (IPComp) is a protocol that improves the performance of communications between hosts by reducing the size of the IP datagrams sent between them. IPComp supports a number of compression algorithms. IxLoad supports the LZ77 algorithm.

IPsec peers can negotiate to use IPComp as part of the setup of a Child SA. A peer requesting a Child SA can advertise that it supports one or more IPComp compression algorithms. The other peer indicates its agreement to use IPComp by selecting one of the offered compression algorithms.

NAT-T

NAT-T (network address translation traversal) was developed to address the problem of using IPsec over NAT devices. NAT devices work by modifying the addresses in the IP header of a packet. Under IPsec, this causes the packets to fail the checksum validation provided by IPsec. To IPsec, the packets appear to have been modified in transit, something IPsec is intended to prevent.

NAT-T detects the presence of NAT devices between two hosts, switches the IPsec function to a non-IPsec port, and encapsulates the IPsec traffic within UDP packets. To preserve the original source and destination port numbers, NAT-T inserts an additional header containing the port numbers between the IP header and the ESP header.

For example, after IKE peers initiate negotiation on port 500, detect support for NAT-T, and detect a NAT device along the path, they can negotiate to switch the IKE and UDP-encapsulated traffic to another port, such as port 4500 (IxLoad listens on port 4500 to establish a connection for IKEv2.).

IKEv2

Version 2 of IKE, defined in RFC 4306, simplifies the IKE protocol. The main differences between IKEv1 and v2 are as follows:

Simplified initial exchange: In IKEv2, the initial contact between peers is accomplished using a single exchange of four messages. IKEv1 provides a choice of eight separate exchange mechanisms.

Reduced setup latency: The initial exchange of two round trips (four messages), coupled with the ability to simultaneously set up a child Security Association (SA) on the back of that exchange, and reduces setup latency for most common setup scenarios.

Fewer header fields and bits: The domain of interpretation (DOI), situation (SIT), and labeled domain Identifier fields have been removed in IKEv2, as have the commit and authentication only bits.

Fewer cryptographic mechanisms: IKEv2 protects its own packets with an ESP-based mechanism very similar to the one it uses to protect IP payloads, simplifying implementation and security analysis.

Increased reliability: In IKEv2, all messages must be acknowledged and sequenced (in IKEv1, message IDs are random), which reduces the number of possible error states.

Resistance to attacks: To better resist attacks, an IKEv2 host does not do much processing until it has satisfied itself that a potential peer is authentic. IKEv1 is vulnerable to DoS attacks (attack by causing excessive processing) and spoofing (access using a forged address).

In addition to the original IKEv2 specification defined in RFC 4306, a subsequent RFC, RFC 4718 IKEv2 Clarifications and Implementation Guidelines, provided further details on implementing IKEv2. IxLoad follows the recommendations in RFC 4718.

Initial Exchanges

Communication between IKEv2 peers begins with exchanges of IKE_SA_INIT and IKE_AUTH messages (in IKEv1, this is known as Phase 1). These initial exchanges normally consist of four messages, although there may be more for some scenarios. All IKEv2 message exchanges consist of request and response pairs.

The first pair of messages (IKE_SA_INIT) negotiate the cryptographic algorithms to be used, exchange nonces, and exchange Diffie-Hellman values.

The second pair of messages (IKE_AUTH) authenticates the previous messages, exchanges identities and certificates, and establishes the first Child SA. Parts of these messages are encrypted and have their integrity protected using keys established through the IKE_SA_INIT exchange, to hide the peers' identities from eavesdroppers. Furthermore, all fields in all messages are authenticated.

Initiator to Responder

The initial exchange begins with the initiator sending the following to the responder:

An IKE header that contains the security parameter indexes (SPIs), version numbers, and has various flags set or unset.

A payload listing the cryptographic algorithms that the initiator supports for the IKE SA.

A payload containing the initiator's Diffie-Hellman shared secret.

A payload containing the initiator's nonce (a random or pseudo-random number that is used only once in a session).

Responder to Initiator

The responder replies to the initiator with the following:

A payload naming the cryptographic suite selected by the responder from those offered by the initiator.

A payload containing the responder's Diffie-Hellman shared secret.

A payload containing the responder's nonce value.

Optionally, the responder may send a certificate request as well.

At this point in the negotiation, each peer uses the nonces and Diffie-Hellman values to generate the seed values to be used in turn to generate all the keys derived for the IKE SA. Keys are generated for encryption and integrity protection (authentication); separate keys are generated for each function in each direction.

An additional value is derived from the Diffie-Hellman values, to be used to generate keys for child SAs.

Beyond this point, all parts of the messages exchanged between the peers are encrypted and authenticated, except for the headers.

Initiator to Responder

In the next series of exchanges, the initiator asserts its identity, proves that it knows the secret corresponding to identity and integrity, and protects the contents of the first message using the AUTH payload. If a certificate was requested, it may return the certificate and a list of its trust anchors. If it does send a certificate, the first certificate provided contains the public key used to verify the AUTH field. At this stage, if the responder hosts multiple identities at the same IP address, the initiator can specify with which of the identities it wants to communicate. The initiator next begins negotiating a child SA.

Responder to Initiator

The responder replies by asserting its own identity, optionally sending one or more certificates (again with the certificate containing the public key used to verify AUTH listed first), authenticates its identity and protects the integrity of the second message with the AUTH payload, and completes negotiation of a Child SA.

Child SAs

What is referred to in IKEv1 as a phase 2 exchange is known in IKEv2 as a Child SA. A Child SA consists of a single request and response pair of messages, and can be initiated by either peer after the initial exchanges are completed.

All messages following the initial exchange are cryptographically protected using the cryptographic algorithms and keys negotiated in the first two messages of the IKE exchange. Because either endpoint can initiate a Child SA, the term initiator in the context of a Child SA exchange refers to the endpoint that initiates the Child SA. The first Child SA is established using messages 3 and 4 of the initial IKE SA exchange, and establishes the parameters for using ESP, AH, and IPComp. Subsequent Child SAs can be initiated to create a new IPsec SA or to perform rekeying of the IKE SA in 2 messages.

Deleting an IKE SA automatically deletes all Child SAs based on it; deleting a Child SA deletes only that Child SA. Unless the Child SA is being used for rekeying, the Child SA exchange includes a Traffic Selector payload. A traffic selector is an address or range of addresses that an IPsec gateway uses to decide what to do with an inbound packet. Traffic Selector payloads specify the selection criteria for packets to be forwarded over SAs.

If an IPsec gateway receives an IP packet that matches a 'protect' selector in its Security Policy Database (SPD), it must protect that packet with IPsec. If there is no SA established, it must create one.

The portion of the Child SA message after the header is encrypted, and the entire message (including the header) is integrity protected (authenticated) using the cryptographic algorithms negotiated for the IKE SA.

Requesting Internal Addresses on Remote Networks

IKEv2 includes a mechanism for external hosts to obtain a temporary IP address for a host on a network protected by a security gateway. This mechanism, described in section 2.19 of RFC 4306, involves adding a Configuration Payload (CP) request to Child SA request.

When a security gateway receives a CP request for an address, it can either obtain an address from an internal pool or it may query external servers (such as DHCP or BOOTP servers) to obtain the address. To return the address, the gateway returns a CP reply.

This mechanism provides IKEv2 with functionality similar to XAUTH and MODE-CFG in IKEv1.

Informational Exchanges

At various points during the life of an IKE SA, the peers may need to send messages to each other regarding control parameters, errors, or notice of certain events. To accomplish this, IKEv2 defines an Informational exchange. Informational exchanges occur only after the initial exchanges and are cryptographically protected with the IKE SA's negotiated keys.

Messages in an information exchange contain zero or more notification, delete, and configuration payloads. An Informational request may contain no payload. In this event, the exchange functions as a Keep Alive message and response.

The recipient of an Informational request always sends a response to it; otherwise, the sender would assume that the message was lost and retransmit it.

Cookies

IKEv1 supported cookies, and IKEv2 continues that support. Internet security association and key management protocol (ISAKMP) fixed message header includes two eight-octet fields titled 'cookies,' and that syntax is used by both IKEv1 and IKEv2, though in IKEv2, they are referred to as the IKE SPI and there is a new separate field in a Notify payload holding the cookie.

Rekeying

Rekeying refers to the re-establishment of SAs to replace SAs that have expired or are about to expire. If attempts to rekey an SA fail, the SA and its Child SAs are terminated. The peers can then negotiate new SAs.

To improve the performance and reduce the potential number of lost packets, most IKE v2 implementations allow SAs to be rekeyed before they expire (in-place rekeying). To rekey a Child SA within an existing SA, a new, equivalent Child SA is created and the old one is deleted. To rekey an SA, a new equivalent SA is created with the peer. The new SA inherits all of the original SA's Child SAs, and the old SA is deleted by sending a message containing a 'Delete' payload over it. The Delete payload is always the last request sent over an SA that terminates normally.

In IKEv1, peers negotiated SA lifetimes with each other. In IKEv2, each peer selects its own lifetime for an SA, and is responsible for rekeying the SA, when necessary. If the two peers select different lifetimes, the peer that selects the shorter lifetime initiates rekeying.

If an SA and its child SAs have carried no traffic for a long time and if its endpoint would not have initiated the SA without any traffic for it, the endpoint may close the SA when its lifetime expires, instead of rekeying it.

Some IKE peers may impose a random delay before initiating rekeying. This is done to prevent a collision-like situation in which both peers select identical lifetimes for an SA, and then simultaneously attempt to rekey it, resulting in duplicate SAs.

IKEv2 does not prohibit duplicate SAs. RFC 4306 states that endpoints can establish multiple SAs between them that have the same traffic selectors to apply different traffic quality of service (QoS) attributes to the SAs.

EAP

In addition to authentication using public key signatures and shared secrets, IKEv2 continues to support the Extensible Authentication Protocol (EAP), which is defined in RFC 3748.

EAP is typically used in scenarios requiring asymmetric authentication, such as users authenticating themselves to a server. For this reason, EAP is typically used to authenticate the initiator to the responder, and in return, the responder authenticates itself to the initiator using a public key signature.

EAP is implemented in IKEv2 as an additional series of AUTH exchanges that must be completed to initialize the IKE SA.

If an initiator wants to use EAP to authenticate itself, it indicates that by omitting the AUTH payload from message 3 of the initial IKE message exchange. Because it has sent an ID payload, but not an AUTH payload, the initiator has declared an identity, but has not proven it. If the responder is willing to allow authentication by EAP, it places an EAP payload in message 4 and defers sending further IKE messages until it has authenticated the initiator in a subsequent AUTH exchange.

IxLoad and IPsec

IxLoad tests a DUT for scalability (the number of tunnels that it can create) and performance (the rate at which it creates them).

This section describes some of the common VPN scenarios and how IxLoad terminology compares with an actual VPN.

Site to Site (Remote Office) Scenario

Figure 46 shows a typical IPsec VPN scenario. In this scenario, an IPsec gateway (DUT) at a corporate location communicates with other IPsec gateways located in remote offices and with roaming users. The IPsec gateways create IPsec tunnels with the central office to protect the data communication between the hosts in the various remote offices. The DUT decrypts the IPsec encrypted traffic that it receives from other gateways and sends the clear text traffic to hosts within the corporate trusted network.

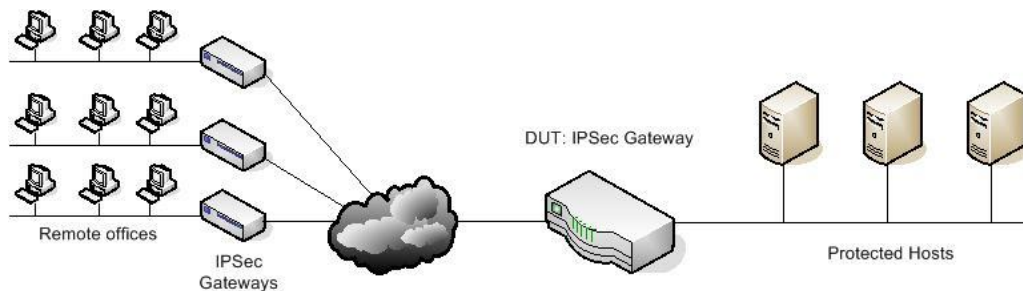


Figure 46. Typical IPsec VPN

Remote Access (Roaming Users) Scenario

Figure 27 Remote access VPN shows another common IPsec VPN scenario: the remote access scenario. In this scenario, the IPsec gateway communicates with remote clients, such as employees who connect to the corporate network using a VPN connection. In this scenario, the roaming clients behave both as the remote IPsec gateways and also as the data source/destination endpoints.

TEST METHODOLOGIES: IPSEC VPN

The emulated remote IPsec gateways are termed Emulated Gateways (EGs) in IxLoad. For the remote-office scenario, the optional emulated subnets behind the remote IPsec gateways are called Emulated Subnets (ESNs). The Ixia ports that emulate EGs and ESNs are referred to as Emulated Gateway ports or Public Side ports.

IxLoad also emulates the hosts on the private network, which are the targets of the phase 2/child SAs established by the EGs. The emulated hosts in the private network that are the data source/destination endpoints are called Protected Hosts (PHs) in IxLoad. The Ixia ports that emulate PHs are called Protected Hosts ports or Private Side ports.

The DUT is assumed to be an IPsec gateway. Two of the DUT's ports are used during testing:

The Public Port is connected to the public, insecure network (emulated by the Emulated Gateways port) and carries IKE communications and IPsec traffic. The address of the interface that is used for establishing IPsec tunnels with the EGs is referred to as the Public Port IP Address.

The Private Port is connected to the private, secure network (emulated by the Protected Hosts port) and carries unencrypted traffic. The address of the interface that is used to forward clear text traffic to the PHs is referred to as the Private Port IP Address.

Figure 49 shows how IxLoad compares with a real VPN.

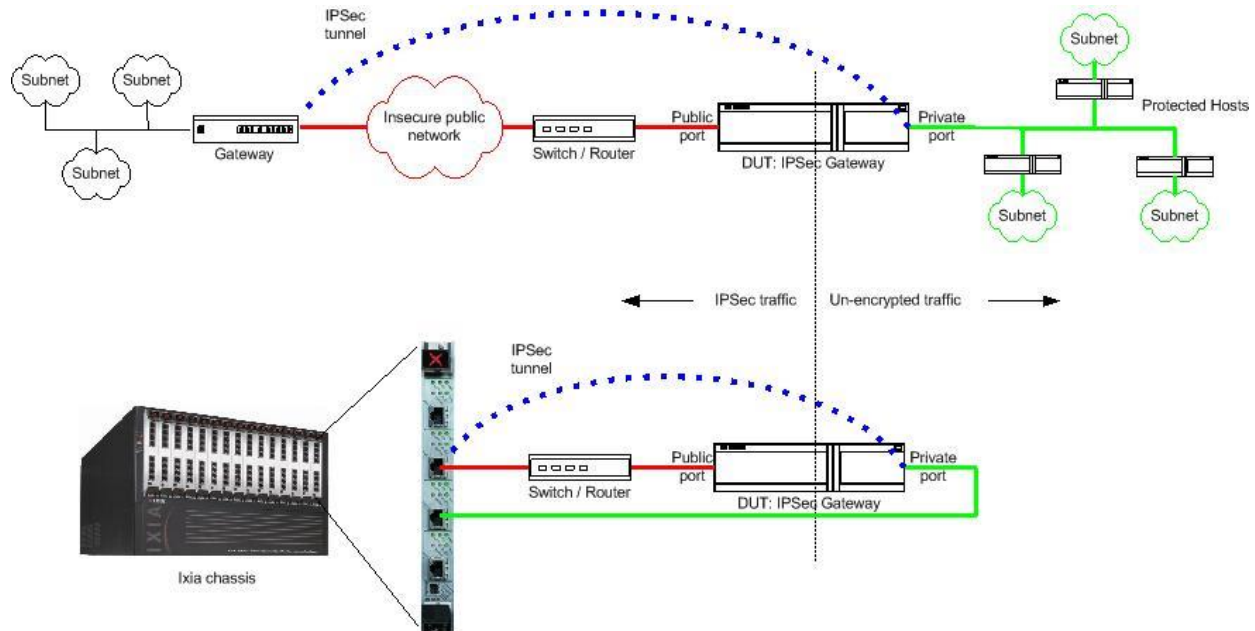


Figure 49. IxLoad versus actual topology

Test Case: IPsec - Data Forwarding Performance

Overview

With the advent of converged networks, security is becoming a prime concern. Being able to support high data rates while having encryption enabled is becoming a popular requirement. High transfer rates must be achieved using small or large packets, or a mix of frame sizes. Depending on the deployment model, data is securely transferred over a small number of IPsec tunnels or across a high number of concurrent tunnels.

IPsec is the most widely used VPN technology. Because it provides protection at the IP level (Layer 3), it can be deployed to secure communication between a pair of gateways, a pair of host computers, or even between a gateway and a host computer. It offers the security features that are required in the enterprise and service provider infrastructures.

Before information can be transferred, an IPsec tunnel is established between two security gateways (SGs) using a two-phase process. Phase 1 establishes communications between the SGs, while Phase 2 establishes the communication for the network behind the SGs. Only after the completion of Phase 2, the tunnel is considered established, and the data sessions between the source and destination hosts can be validated.

Because of the protocol complexity, IPsec performance can have degradations due to a large diversity of factors. Depending on the architecture of every DUT, its capacity to encrypt and decrypt traffic may be more or less impacted.

Data rates performance is primarily affected by the following factors:

- Encryption algorithm type (DES/3DES/AES) and its key length (128, 192, 256 bit)

 - DES (56 bit) and 3DES (168 bit)

 - AES128 (128 bit), AES192 (192 bit) and AES256 (256 bit)

 - Null

TEST CASE: IPSEC – DATA FORWARDING PERFORMANCE

A sample result highlighting the throughput performance of a security gateway while using different encryption algorithms is shown in the following figure:

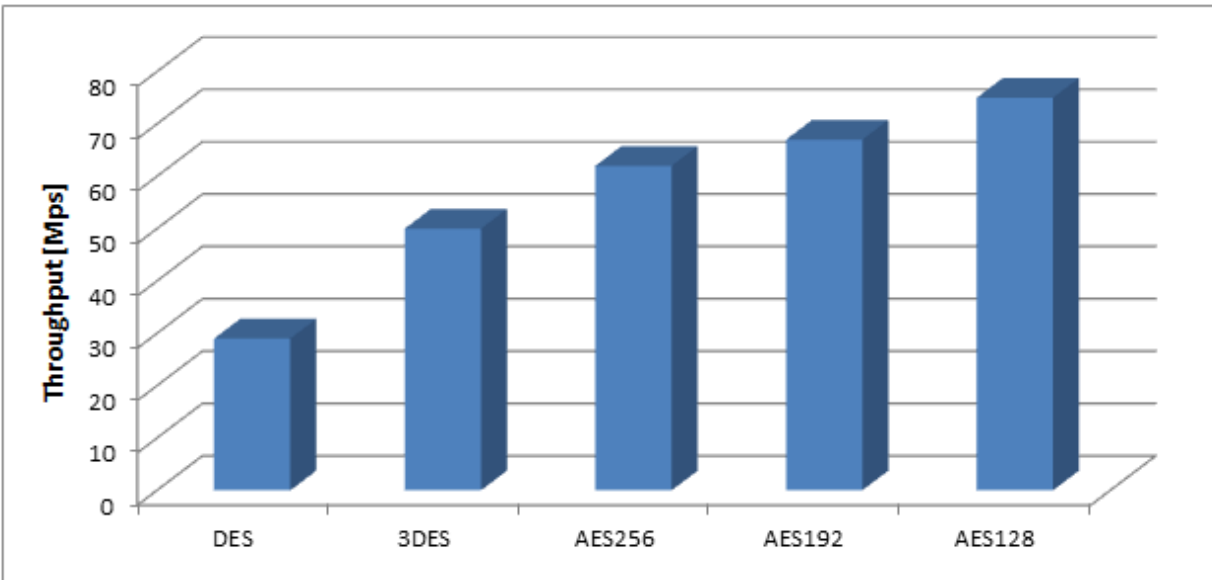


Figure 50. Data rate performance vs. encryption algorithm

Key size of the encryption algorithm

For example: 128 bit, 192 bit or 256 bit for AES encryption (see sample results in the preceding figure)

Hash Algorithm type (HMAC-MD5 vs. HMAC-SHA1)

HMAC-MD5 is expected to have a better performance compared with HMAC-SHA1 because of the size of the secret key, which is 128 bytes compared with 160 bytes for SHA1

The traffic type

small packets versus large packets versus IMIX
UDP versus TCP
stateless versus stateful
data traffic versus voice traffic

Number of concurrent tunnels that are concurrently used to exchange traffic

The overall IPsec rekey rate

Rekeying may degrade performance by increasing frame loss because of tunnel renegotiation

The overall Dead Peer Detection rate

TEST CASE: IPSEC – DATA FORWARDING PERFORMANCE

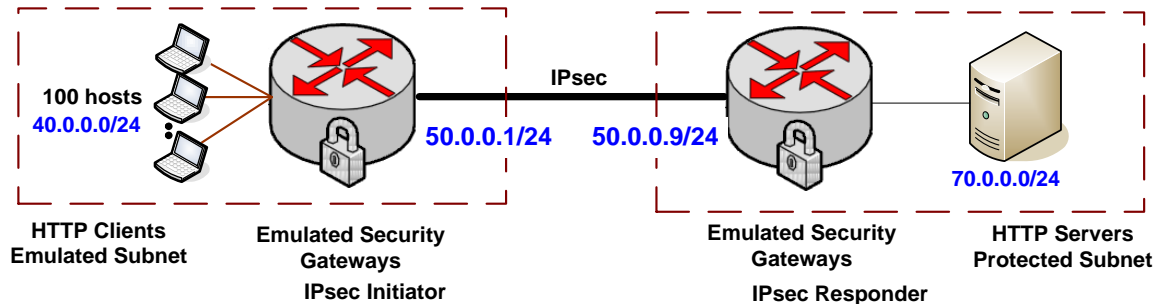
Objective

This test methodology provides step-by-step instructions that demonstrate how to configure IxLoad to measure the maximum amount of statefull HTTP traffic that can be securely exchanged over 100 IPsec tunnels by using a user defined IMIX traffic pattern.

Setup

The test setup consists of a pair of Accelaron NP ports connected back to back. Each Ixia Accelaron port emulates four security gateways, each one with 100 hosts behind.

On the emulated subnets, IxLoad is configured to emulate HTTP clients, and on the protected subnets, IxLoad is configured to emulate HTTP servers. We use the HTTP Throughput objective to determine the maximum data rates that can be forwarded between Ixia's ports.



Step-by-Step Testing

Defining the Network and Traffic Flows

1. Create two Networks (Network1 and Network2).
2. To Network1, add an HTTP Client activity.
3. To Network2, add an HTTP Server activity.

TEST CASE: IPSEC – DATA FORWARDING PERFORMANCE

- Using the 'lollipop' connector exposed on the right side of the HTTPClient1 activity, drag a symbolic link over HTTPServer1. This creates a basic HTTP configuration for back-2-back. The result is as follows:

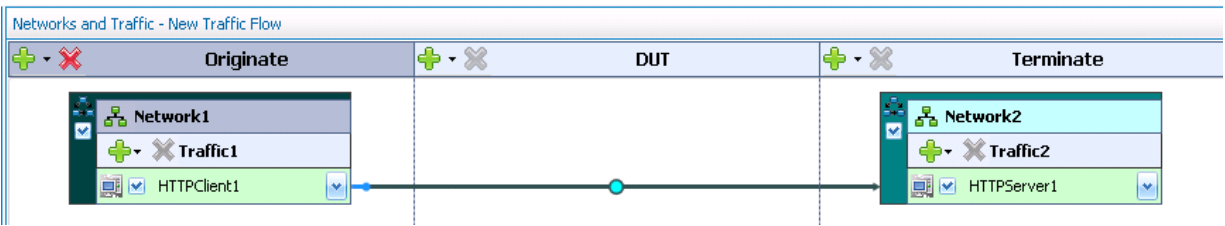


Figure 51. Overview of Network and Traffic Flow

Note: To achieve the maximum throughput performance, select the activity HTTPClient1 and change the HTTP page size to /1024k.html (see the parameters of GET 1 command listed under Commands page).

Property Name	Property Value
Destination(IP or IP:Port)	Traffic2_HTTPServer1:80
Page/Object	/1024k.html
Abort	None
NameValueArgs	
Profile	None

Figure 52. Command Editor - preview of the HTTP GET settings

By default, the throughput objective seeks for the optimal number of users and IPs to generate the maximum data rates. This may result, however, in a small number of IPs generating traffic. You can enforce all the IPs to generate data traffic by setting Cycle users through all source IPs for the HTTPClient1 activity. To do so, select Traffic1, select the IP Mappings page, and then set the option from the list.

Activity	User Source IP Rule (per port)
HTTPClient1	Cycle Users Through All Source IPs

Network Ranges By Port Distribution Group	Activities & Endpoints
	HTTPClient1
	HTTP

Figure 53. User source IP mapping

TEST CASE: IPSEC – DATA FORWARDING PERFORMANCE

As we will see later, the Throughput objective allows a constraint of simulated users, which can force all users or only a subset to generate traffic. When the number of users applied as a constraint is smaller than the number of IPs defined at the network level, the 'Cycle Users Through All Source IPs' forces the emulated HTTP Clients to use all the IPs by cycling through all the available IPs.

IPsec configuration using the IPsec Network Wizard

To simplify IPsec configuration, IxLoad provides an IPsec wizard. To start the **IPsec Network wizard**, select the **Wizards menu > | New IPsec Network Wizard**.

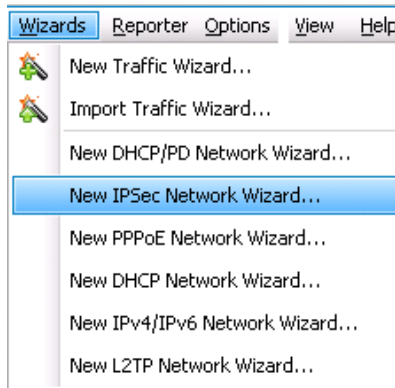


Figure 54. Launch the IPsec Network Wizard:

1. At the first configuration step of the Wizard, set the following:

Test Type: Port-to-Port

Test Scenario: Site-to-Site

IKE Version: IKEv1

Number of IPsec tunnels per Range = 1

This setting controls the number of Emulated IPsec Gateways (the host parameter under IP network stack).

TEST CASE: IPSEC – DATA FORWARDING PERFORMANCE

Number of IP/IPsec Ranges per Network Group = 1

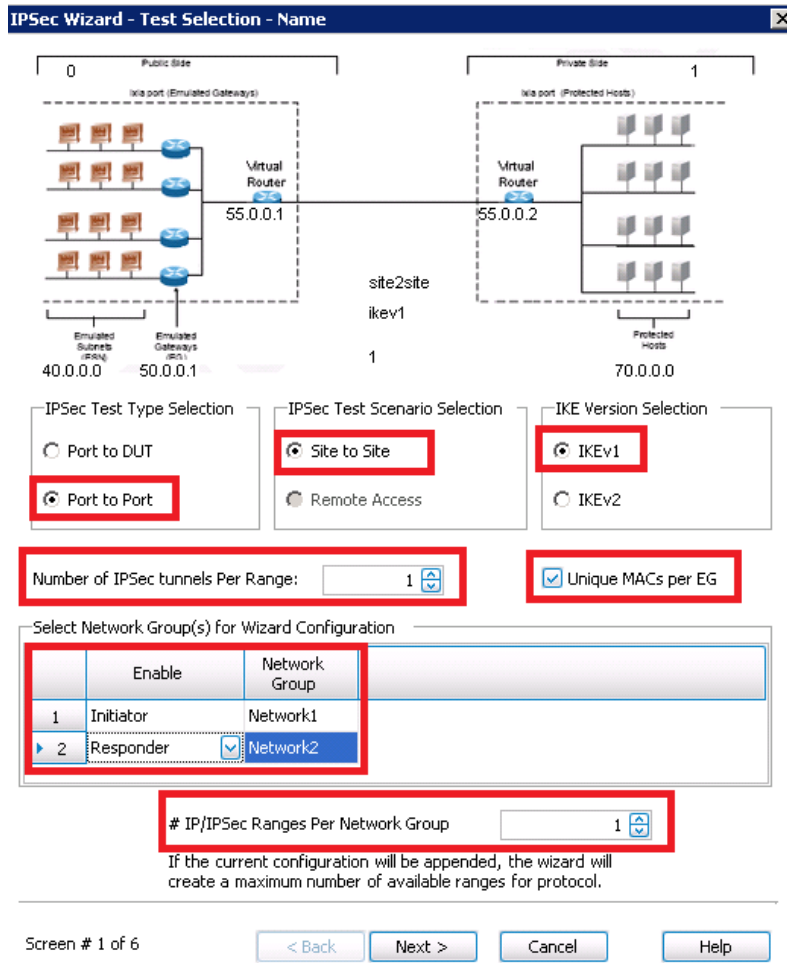


Figure 55. IPsec Network Wizard - screen #1 of 6

Network1 as IPsec Initiator

Network2 as IPsec Responder

Unique MAC per EG⁹: Checked

⁹ Selecting 'Unique MAC per EG' will position the IP network stack directly over the MAC network stack. Clearing the Unique MAC per EG option will result in an intermediate Emulated Router network stack added between the IP and MAC network stacks. When Emulated Router stack is added, all IPsec Emulated Gateways are placed behind a Virtual Router, which allows a single MAC address and VLAN tag to be configured. Removing the MAC layer allows configuration of unique MAC and VLAN per Emulated Gateway.

TEST CASE: IPSEC – DATA FORWARDING PERFORMANCE

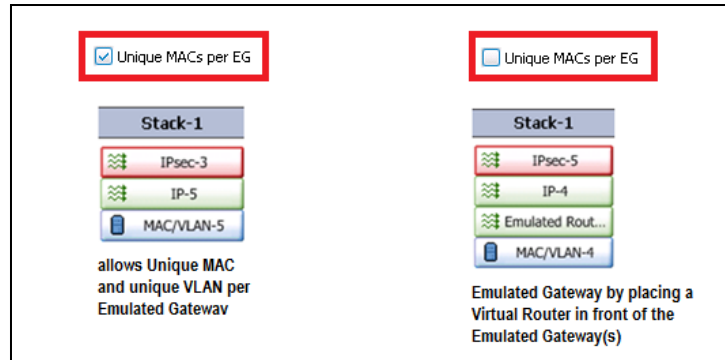


Figure 56. Comparison of the results of Unique MACs per EG

- At the second configuration step of the Wizard, set the following:

Phase1 Hash Algorithm: HMAC-SHA1

Phase1 Encryption:

AES-128

Phase2 Hash Algorithm: HMAC-SHA1

Phase2 Encryption:

AES-128

Leave the other options to their default values:

IKE Mode = Main Mode

AH&ESP = ESP

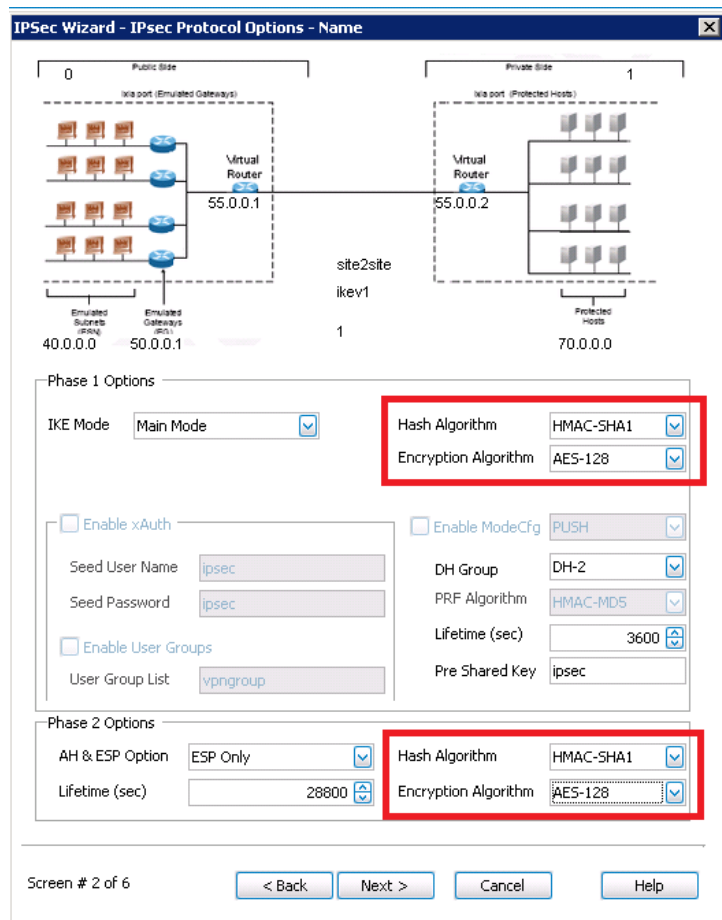
DH Group = DH-2

PreShared Key = IPsec

Phase1 Lifetime 3600 (sec)

Phase2 Lifetime 28800 (sec)

- Select **Next** to continue.



TEST CASE: IPSEC – DATA FORWARDING PERFORMANCE

- At the third screen of the IPsec Network Wizard, do the following:

Set the number of hosts on the Emulated Subnets to 100.

Set the number of hosts on the Protected Subnet to 100.

Note: This setting will place 100 hosts behind the Emulated IPsec Gateway.

Keep the remaining settings to their default values as highlighted in the following figure.

IPsec Wizard - IPsec Network Options - Name

Public Side (1) Private Side (1)

Virtual Router 55.0.0.1 Virtual Router 55.0.0.2

site2site ikev1 100

Emulated Subnets (40.0.0.0) Emulated Gateways (50.0.0.1) Protected Hosts (70.0.0.0)

Subnets IP Address Type IPv4 Gateways IP Address Type IPv4

Public Area Configuration

Emulated Gateways

First IP Address 50.0.0.1

Increment By 0.0.0.1

Range Increment Step 0.0.1.0

Prefix 24

Emulated Subnets

First IP Address 40.0.0.0

Increment By 0.0.1.0

Range Increment Step 0.1.0.0

Num Hosts 100

Prefix 24

Private Area Configuration

Emulated Gateways

First IP Address 60.0.0.1

Increment By 0.0.0.1

Range Increment Step 0.0.1.0

Prefix 24

Protected Subnets

First IP Address 70.0.0.0

Increment By 0.0.1.0

Range Increment Step 0.1.0.0

Num Hosts 100

Prefix 24

Single Protected Subnet

DUT Configuration

Public IP Address Private IP Address

Screen # 3 of 6

< Back Next > Cancel Help

Figure 57. IPsec Network Wizard, configuration screen 3

- Select Next to continue.
- At wizard's page number 4 and screen number 5 of the IPsec Network Wizard, keep all settings to their default values

TEST CASE: IPSEC – DATA FORWARDING PERFORMANCE

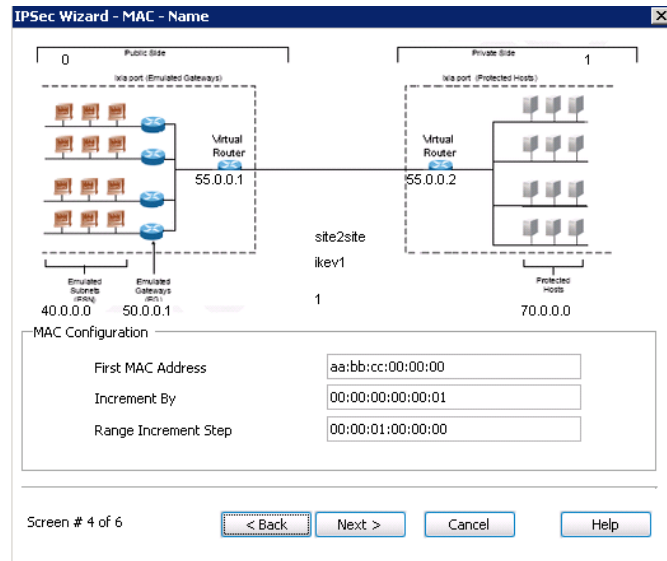


Figure 58. Wizard's screen number 4: MAC configuration

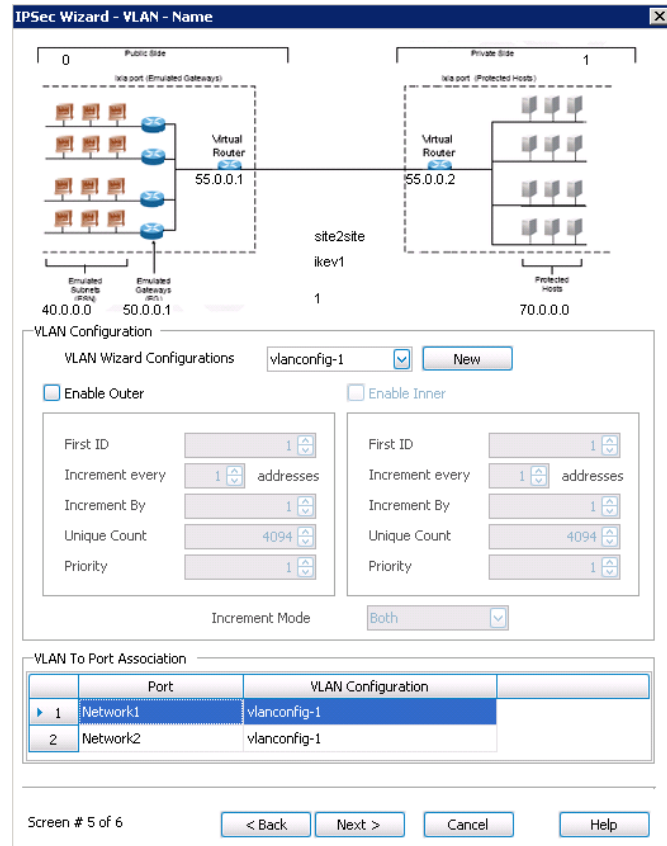


Figure 59. VLAN configuration

7. Click **Generate and Overwrite Existing Configuration**, and then click **Finish** to apply your configuration

TEST CASE: IPSEC – DATA FORWARDING PERFORMANCE

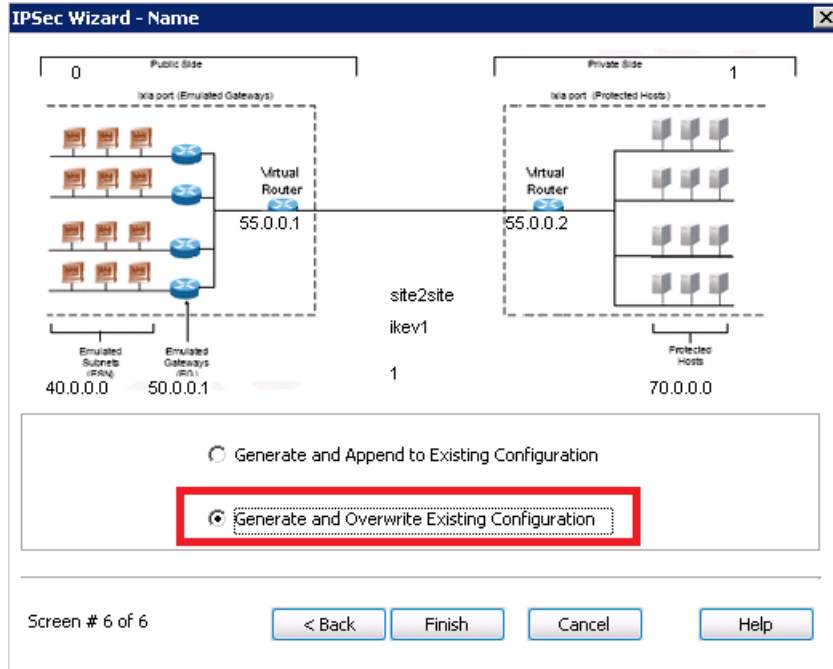


Figure 60. Generate and overwrite Existing Configuration

Review and validate the configuration generated by the IPsec Network Wizard

Take a moment to review and understand how the IPsec Network Wizard configured the parameters of IP and IPsec network stacks available under Network1 and Network2 objects.

Pay special attention to the following fields:

1. Under IP network stack, review:

The IP address field -> defines the address of the Emulated IPsec Gateway.

The IP Count field -> defines the number of Emulated IPsec Gateways.

2. Under IPsec network stack, review:

Network Config page

Note the host count = 100 (emulates 100 hosts behind the Emulated Gateway)

Note the subnet set for the Emulated Subnet.

Note the subnet set for the Protect Subnet.

Note the Public Peer IP address.

Basic configuration page

Authentication configuration page

IKE configuration page

TEST CASE: IPSEC – DATA FORWARDING PERFORMANCE

Identification configuration page

IKE Phase 2 configuration page

IKE Control configuration page

3. Review the IPsec role of each network by selecting the IPsec network stack, and then the **Network Settings** page. As per our settings, Network1 must act as IPsec Initiator and Network2 must act as IPsec Responder.

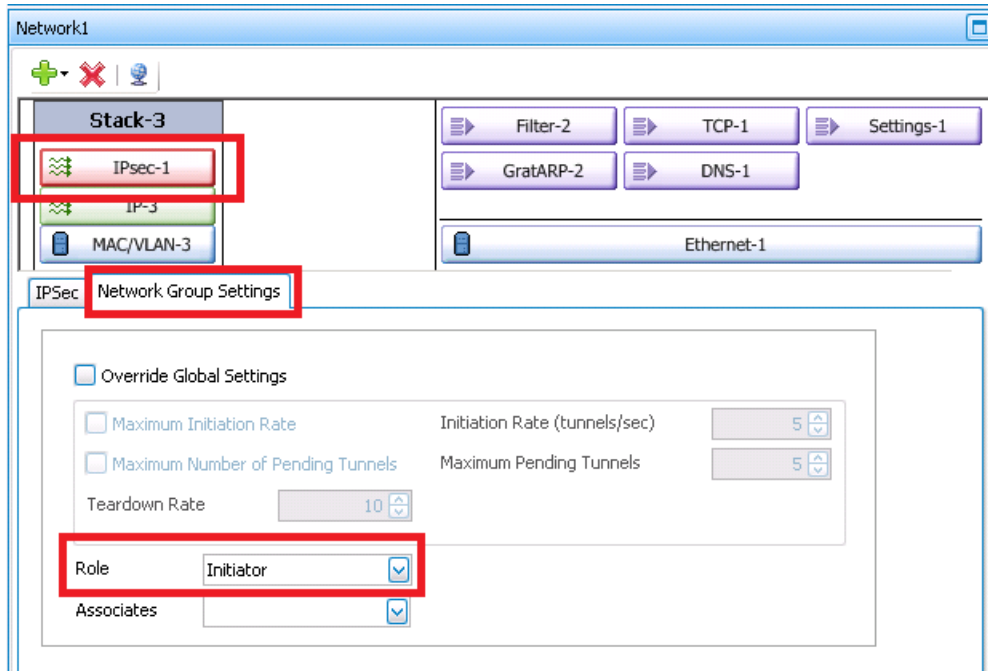


Figure 61. IPsec Role (Initiator or Responder)

Setting the INITIAL_CONTACT Notification Message

1. Select the IPsec stack of Network1.
2. Select the IKE Control page.
3. Select the check box from the Initial Contact field.

TEST CASE: IPSEC – DATA FORWARDING PERFORMANCE

4. Select **Network2 | IPsec network stack | IKE Control page** and select the **Initial Contact** check box.

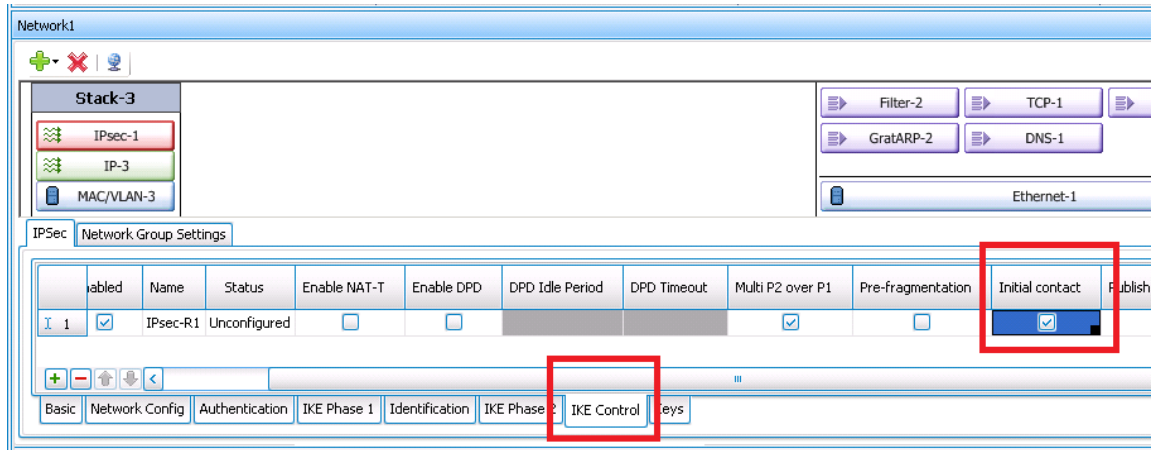


Figure 62. IKE Control page - enabling Initial Contact

Notes:

When you select the **Initial Contact** check box, the IPsec plug-in sends the INITIAL_CONTACT notification payload as part of IKE SA establishment. Note that IxLoad's IPsec plug-in always ignores the INITIAL_CONTACT notification payload, if it is received. By default, this parameter is disabled.

Setting Multiple Phase2 over Phase1

Before data plane traffic can be transferred, a 'tunnel' is created between two security gateways by using a two-phase process. Phase 1 establishes communications between the security gateways, while Phase 2 establishes the communication for the network behind the security gateways.

1. Select the IPsec stack of Network1.
2. Select the **IKE Control** page.
3. Select the **Multiple P2 over P1 (Multiple Phase 2 over Phase1)**.

TEST CASE: IPSEC – DATA FORWARDING PERFORMANCE

4. Select **Network2 | IPsec network stack | IKE Control** page and select the **Multiple P2 over P1** check box.

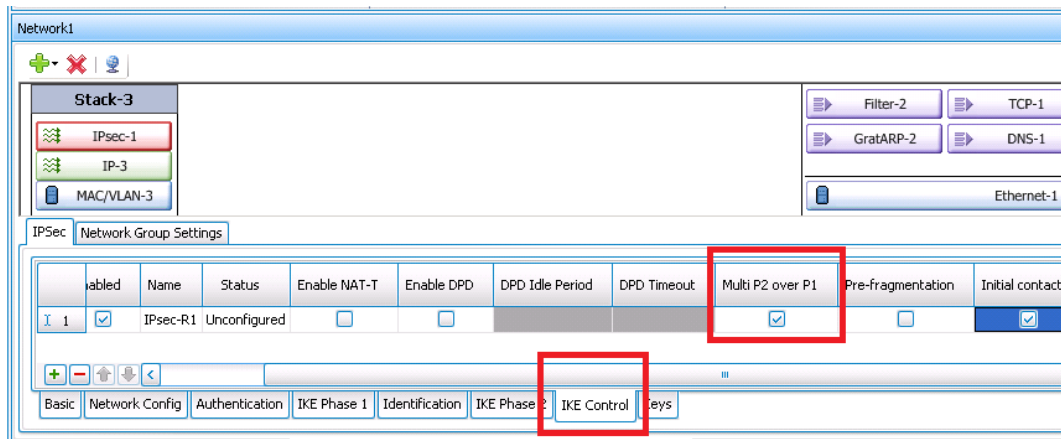



Figure 63. Enabling Multiple Phase2 over Phase1

The number of Phase1 and Phase2 tunnels that are established depends on the number of emulated gateways and the number of ESN hosts in the emulated subnet. Our configuration includes a pair of Emulated Gateways (50.0.0.1 and 50.0.0.2) and 100 ESN hosts. Hence, the configuration connects 1 Phase1 tunnel and 100 Phase 2 tunnels.

Configuring the tunnel setup and tunnel teardown rates

1. Select either **Network1** or **Network2**.
2. Click **Network Plug-in Global Settings**  button.

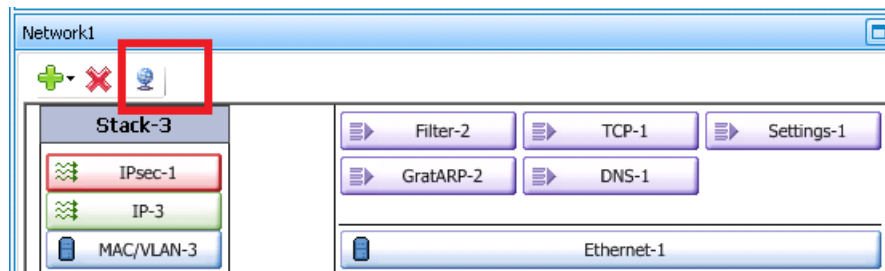


Figure 64. Network - Global Settings

TEST CASE: IPSEC – DATA FORWARDING PERFORMANCE

3. The **Network Plug-In Settings** window is displayed.

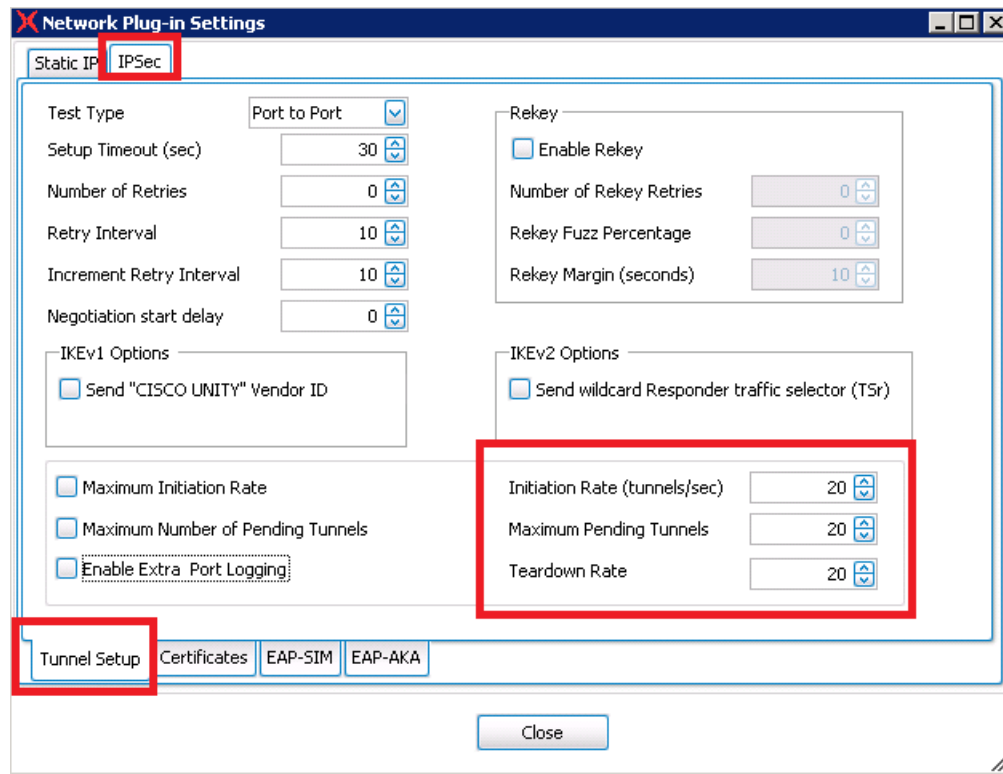


Figure 65. Global IPsec Settings

4. Select the **IPsec** page | **Tunnel Setup** page.
5. Change the following global parameters:
 - Set Initiation Rate = 20
 - Set Maximum Pending Tunnels = 20
 - Set Maximum Teardown Tunnels = 20

Test Objective

1. Select the **Timeline & Objective** step.
2. Select the **HTTPClient1** activity to set its objective.
3. Set the test objective as **Throughput (Kbps)** with a value of 100,000 Kbps.
4. Select the **Timeline** configuration page.
5. Set the **Sustain Time** to **3 minutes**.

TEST CASE: IPSEC – DATA FORWARDING PERFORMANCE

- Set the **Ramp Down Time** to 10 seconds.

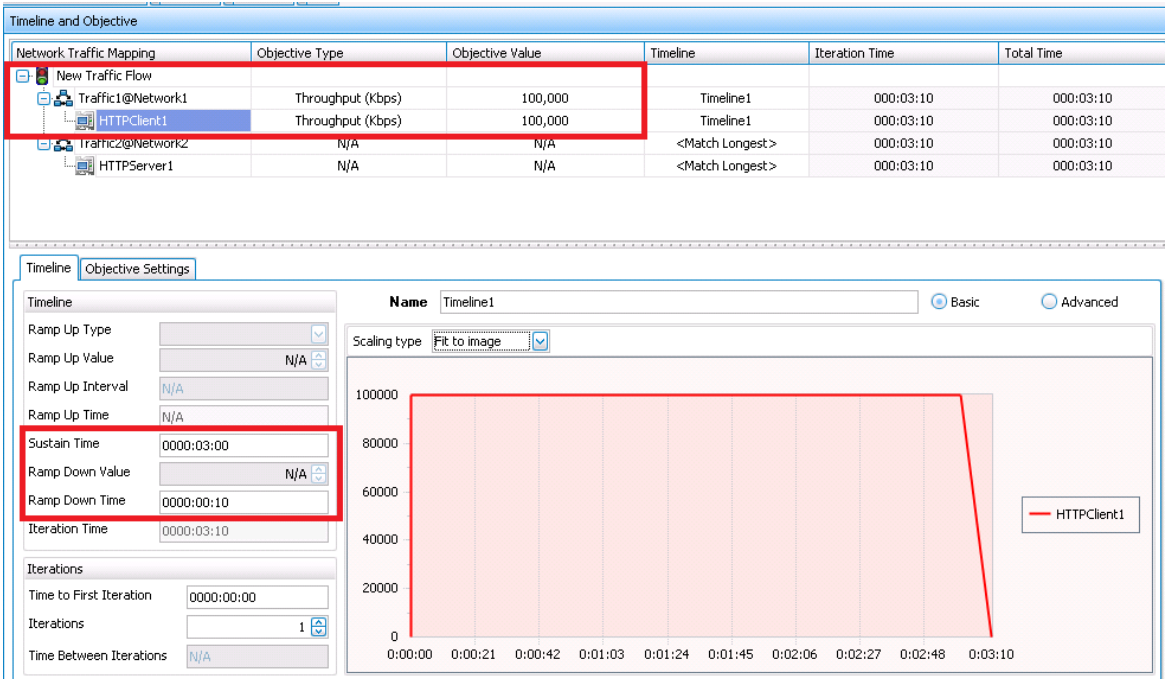


Figure 66. Setting test objective

- Select the **Objective Settings** page.
- Set the simulated users constraint to 25.
- Set the ramp up value to 25 users.

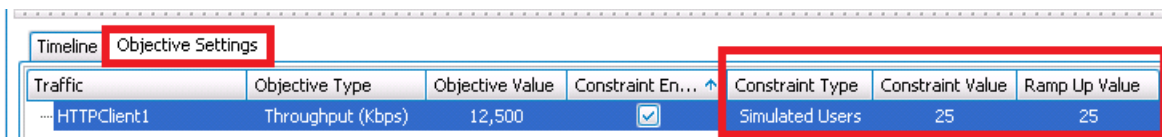


Figure 67. Objective constraint settings

This setting forces all the IPsec tunnels to generate traffic by cycling users through all IPs using groups of 25 simultaneous users. The 'Cycle users through all source IP' option will have no effect if the simulated user's constraint is set to 100 (matching the number of IP addresses configured).

Ports Assignment

The test setup requires two Accelaron NP ports connected back-2-back (1GE Non Aggregated Mode).

- From the **Test Configuration** panel, click **Port Assignments**.
- Add your chassis by clicking **Add Chassis**.
- Assign one port to each network.

Test Options

1. Click **Test Options** and set the following parameters:

Forcefully Take Ownership

Reboot Ports before Configuring

Release Configuration after Test

CSV Polling Interval: 2 [seconds]

Enable Network Diagnostics

Enable Show Diagnostics from Apply Config

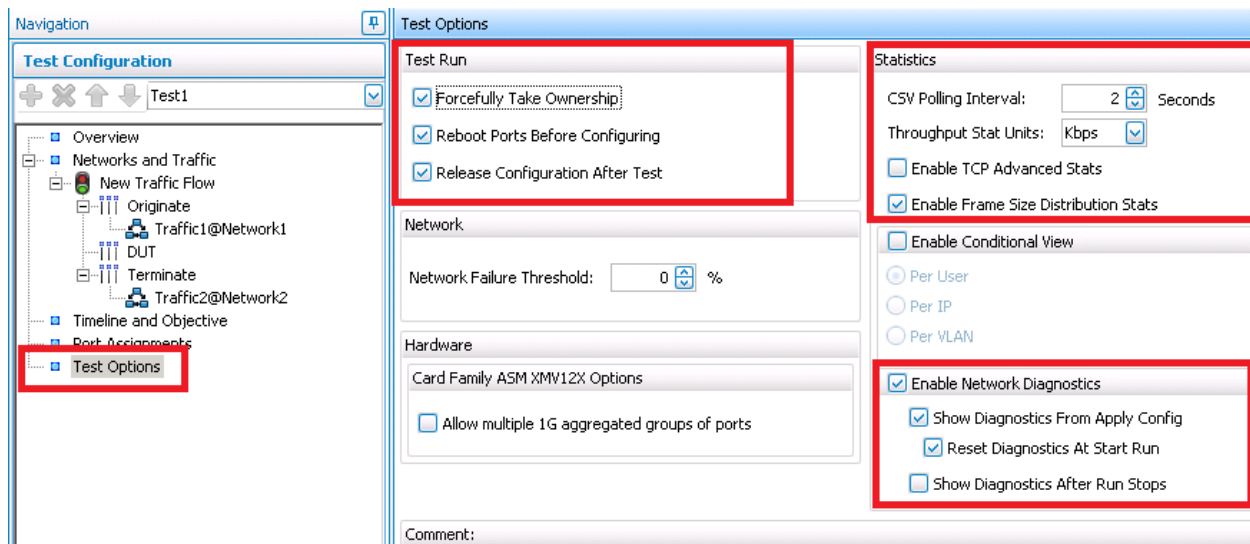


Figure 68. Test Options

Note: To enable the IPsec stats, you must select the **Show Diagnostics From Apply Config** check box.

2. Save your configuration by clicking **File | Save**.
3. Run the test by clicking **Test | Start** or by clicking the **Start** button on the toolbar.

TEST CASE: IPSEC – DATA FORWARDING PERFORMANCE

Test Variables

The main test variables impacting the data rate performance are as follows:

Parameter Name	Current Value	Comments
MSS Value	1460 bytes	<p>Available options</p> <p>MSS value is user configurable (see IP stack).</p> <p>Data Rate Performance</p> <p>Higher degradation for smaller MSS values.</p> <p>Recommended Trials</p> <p>Repeat the test for the following MSS values: 64, 128, 256, 512, and 1024 bytes.</p>
Phase 1 & 2 Encryption Algorithm	AES128	<p>Available options:</p> <p>Null, DES, 3DES, AES 128/192/256</p> <p>Data Rate Performance</p> <p>Repeat the test for DES, 3DES, and AES256.</p> <p>Recommended Trials</p> <p>Repeat the test for Null, DES, 3DES, and AES256,</p>
Phase 1 & Phase 2 Hash Algorithm	HMAC-SHA1	<p>Available options:</p> <p>MD5 & SHA1</p> <p>Data Rate Performance</p> <p>HMAC-MD5 is expected to have a better performance than HMAC-SHA1 because of the size of the secret key.</p> <p>Recommended Trials</p> <p>Repeat the test for Null, DES, 3DES, and AES256.</p>

TEST CASE: IPSEC – DATA FORWARDING PERFORMANCE

Parameter Name	Current Value	Comments
HTML page size on HTTP GET command	./1024K.html	<p>Available options:</p> <p>User configurable</p> <p>Data Rate Performance</p> <p>Higher degradation for smaller HTTP pages as is translating in smaller packets.</p> <p>Recommended Trials</p> <p>Repeat the test for ./8k.html.</p>
Traffic Type	Statefull HTTP	<p>Available options:</p> <p>Stateless UDP, TCP</p> <p>Statefull traffic – all traffic types supported by IxLoad</p> <p>Data Rate Performance</p> <p>Higher degradation when stateful traffic is used.</p> <p>Recommended Trials</p> <p>Trials using other L4-7 activities (stateless and statefull), including application mix.</p>
Tunnel Flapping Dynamic Control Plane mode	OFF	<p>Available options:</p> <p>On and Off</p> <p>Data Rate Performance</p> <p>Higher degradation when tunnel flapping is enabled because control plane and data plane may share same CPU.</p> <p>Recommended Trials</p> <p>ON</p>

Dynamic Control Plane and Tunnel Flapping

The **Settings** button allows you to enable or disable dynamic control plane mode. The **Settings** window is shown in the following figure, and allows three different modes:

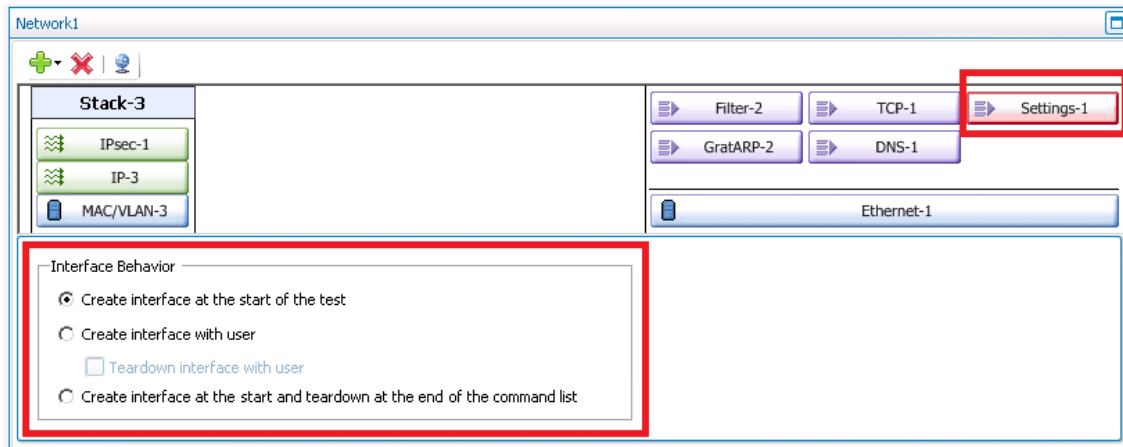


Figure 69. Dynamic Control Plane

The first option—**Create interface at the start of the test**—disables dynamic control plane mode, and enables the traditional IxLoad behavior: all interfaces are negotiated during the Apply Config time, when the test configuration is applied to the Ixia ports. If you run IPsec in this mode, the application waits for all the tunnels to connect, and only after successful establishment of all the tunnels, the L4-7 activity traffic will start to configure and to transmit traffic.

The second option—**Create interface with user**—enables Dynamic Control Plane. If you select this mode, all interfaces are negotiated after the test is started, at the same time as the users are created. When IPsec is used, this results in transmission of data plane traffic immediately after the tunnel was established.

The **Teardown interface with user** option allows interfaces to be torn down in the ramp-down stage along with the users, assuming that the users terminate gracefully during the ramp-down period. The sessions for any users that cannot terminate gracefully during ramp-down are forcefully torn down when the test stops.

If this option is not selected, the interfaces are torn down after the test stops. For the next test run, the sessions are negotiated again.

The third option—**Create interface at the start and teardown at the end of the command list**—enables Dynamic Control Plane. If you select this mode, all interfaces are negotiated when the command list begins execution and Interfaces are torn down after the command list has been completed. When used with IPsec, this will result in IPsec tunnels being established, and then torn down with every loop of the command list.

Results Analysis

This section reviews the statistics important for our test:

TEST CASE: IPSEC – DATA FORWARDING PERFORMANCE

L2 and HTTP throughput

Select the **L2-3 Throughput** statistics view to monitor the L2 TX and RX rate provided individually for both client and server sides.

Select the **HTTP – Throughput Objective** statistics view to monitor the HTTP Throughput.

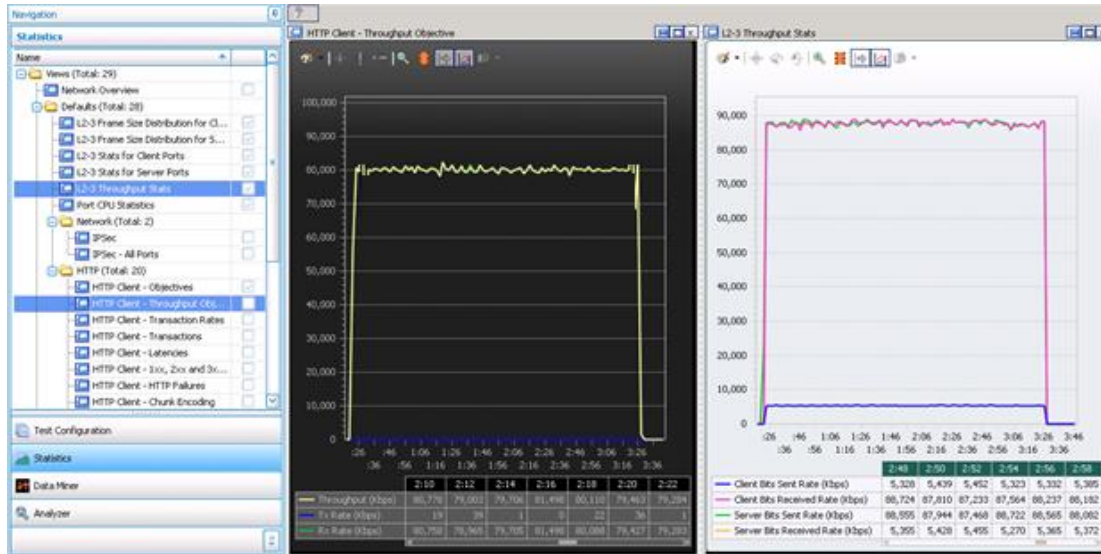
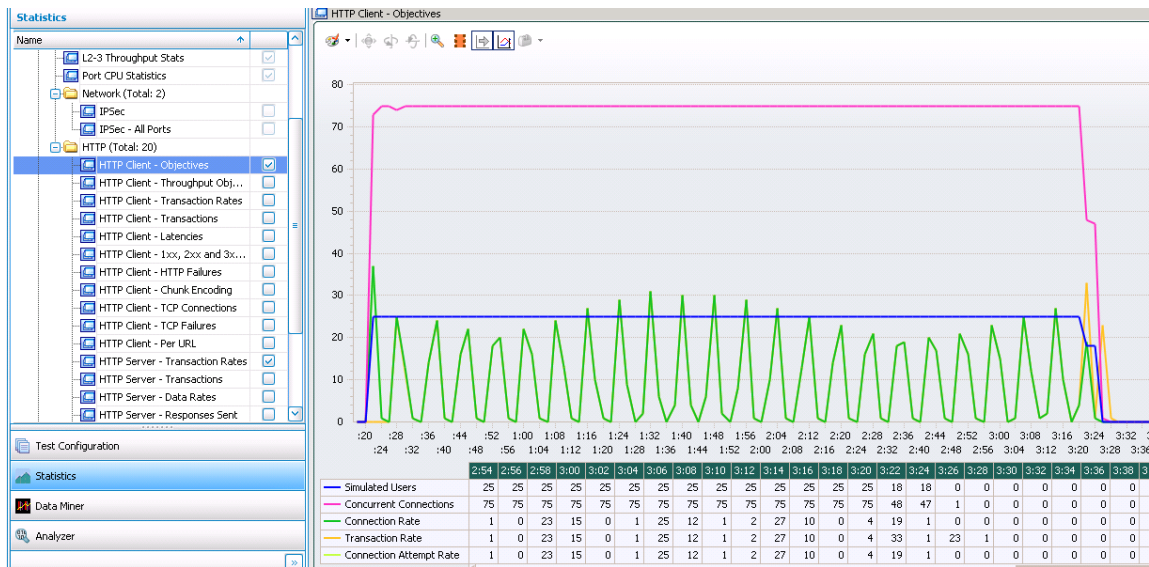


Figure 70. HTTP Throughput versus L2 throughput

Select the **HTTP Client – Objectives** statistics and check the number of simulated HTTP users.



Select the **IPsec – All Ports** statistics view to confirm that the following:

100 IPsec tunnels are initiated.

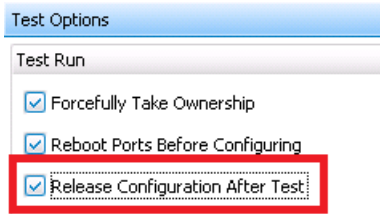
100 IPsec tunnels are established.

TEST CASE: IPSEC – DATA FORWARDING PERFORMANCE

100 IPsec tunnels are active for the entire sustain time.

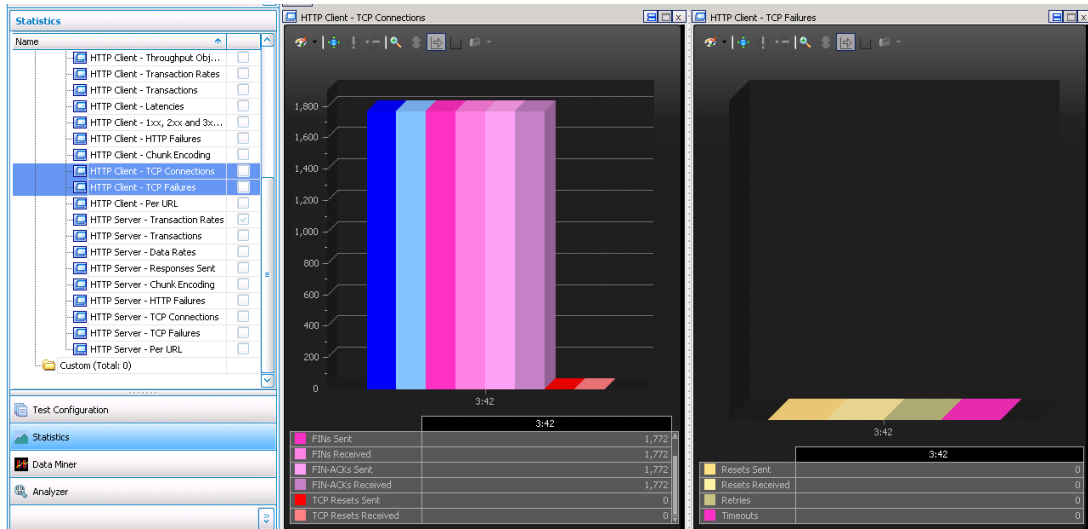
Stat Name	04	3:06	3:08	3:10	3:12	3:14	3:16	3:18	3:20
IPSec/All Ports : Sessions Initiated	100	100	100	100	100	100	100	100	100
IPSec/All Ports : Sessions Succeeded	100	100	100	100	100	100	100	100	100
IPSec/All Ports : Sessions Failed	0	0	0	0	0	0	0	0	0
IPSec/All Ports : Sessions Active	100	100	100	100	100	100	100	100	100
IPSec/All Ports : Initiated Tunnel Rate	0	0	0	0	0	0	0	0	0
IPSec/All Ports : Setup Tunnel Rate	0	0	0	0	0	0	0	0	0
IPSec/All Ports : Failed Tunnel Rate	0	0	0	0	0	0	0	0	0
IPSec/All Ports : Total Retries	0	0	0	0	0	0	0	0	0
IPSec/All Ports : Phase1 Failed Rekeys	0	0	0	0	0	0	0	0	0
IPSec/All Ports : Phase1 Successful Rekeys	0	0	0	0	0	0	0	0	0
IPSec/All Ports : Phase1 Total Rekeys	0	0	0	0	0	0	0	0	0
IPSec/All Ports : Phase2 Failed Rekeys	0	0	0	0	0	0	0	0	0
IPSec/All Ports : Phase2 Successful Rekeys	0	0	0	0	0	0	0	0	0
IPSec/All Ports : Phase2 Total Rekeys	0	0	0	0	0	0	0	0	0
IPSec/All Ports : # DPD Hellos sent	0	0	0	0	0	0	0	0	0
IPSec/All Ports : # DPD Hellos received	0	0	0	0	0	0	0	0	0
IPSec/All Ports : # DPD ACKs received	0	0	0	0	0	0	0	0	0
IPSec/All Ports : Interfaces up	0	0	0	0	0	0	0	0	0
IPSec/All Ports : Interfaces down	0	0	0	0	0	0	0	0	0
IPSec/All Ports : Interfaces failed	0	0	0	0	0	0	0	0	0
IPSec/All Ports : Interfaces outstanding	0	0	0	0	0	0	0	0	0

Note: At the end of the test, the tunnels are disconnected (torn-down) only if you had selected the **Release Configuration After Test** option under **Test Options**.

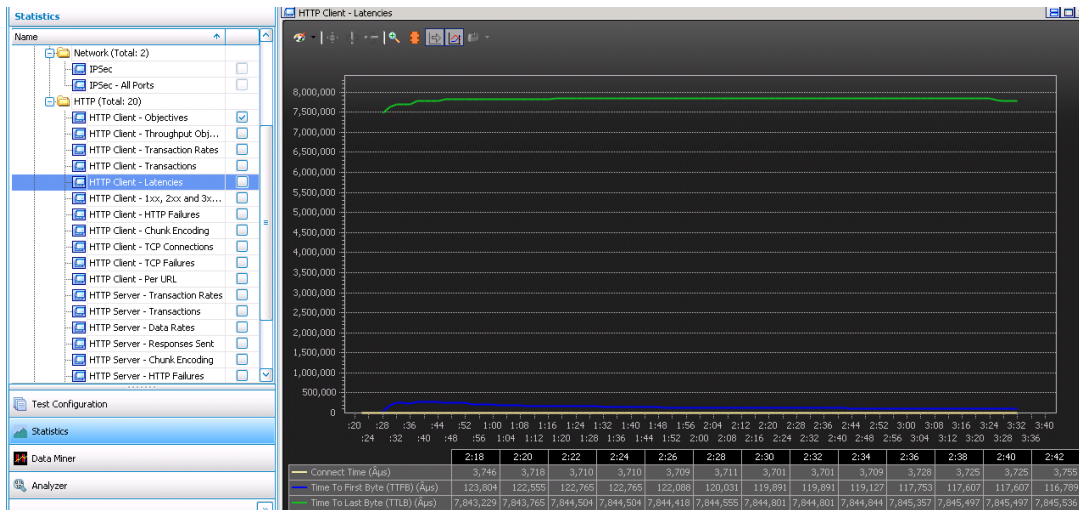


TEST CASE: IPSEC – DATA FORWARDING PERFORMANCE

Use the **HTTP Client – TCP Connections** and **HTTP Client – TCP Failures** statistics view to determine if any TCP Resets were sent or received, the number of TCP timeouts, and retries.



Use the **HTTP Client – Latencies** view to assess the Connect Time (μ s), Time to First Byte (μ s), and Time to Last Byte (μ s),



Conclusions

This test methodology demonstrates how to configure IxLoad to determine the maximum HTTP data rates that can be securely transmitted over 100 IPsec tunnels and reviewed some of the main parameters that affects the data rate performance.

Test Case: IPsec - Tunnel Capacity Test

Overview

IPsec is the most widely used VPN technology. Because it provides protection at the IP level (Layer 3), it can be deployed to secure communication between a pair of gateways, a pair of host computers, or even between a gateway and a host computer.

The tunnel capacity depends directly on the available memory. For the same amount of memory, the following settings can lead to higher memory consumption:

Authentication Method:

PreShared Key

RSA Certificates

EAP (SIM, AKA, TLS, and MD5)

Note: PreShared Key consumes less memory than Certificates and EAP based authentication.

IPv6 versus IPv4: More memory is required for IPv6 based endpoints.

The number of hosts per IPsec tunnel: The memory per tunnel increases with the number of hosts configured per tunnel.

Objective

This test methodology provides systematic instructions that demonstrate how to configure IxLoad to measure the maximum number of IPsec tunnels that can be established between two Ixia Acceleron NP ports. The test also validates that the tunnel can transfer data sessions immediately after establishment and at end of the test.

TEST CASE: IPSEC – TUNNEL CAPACITY TEST

Setup

The test setup consists in a pair of Acceleron NP ports connected back to back. IxLoad emulates 30,000 Security Gateways on each side. Each SGW has a single host behind. On successful establishment of all the tunnels, we generate HTTP traffic at a configured data rate. We use the HTTP Throughput objective to control the data rate and the number of simultaneous users that are sending traffic between Ixia's ports.

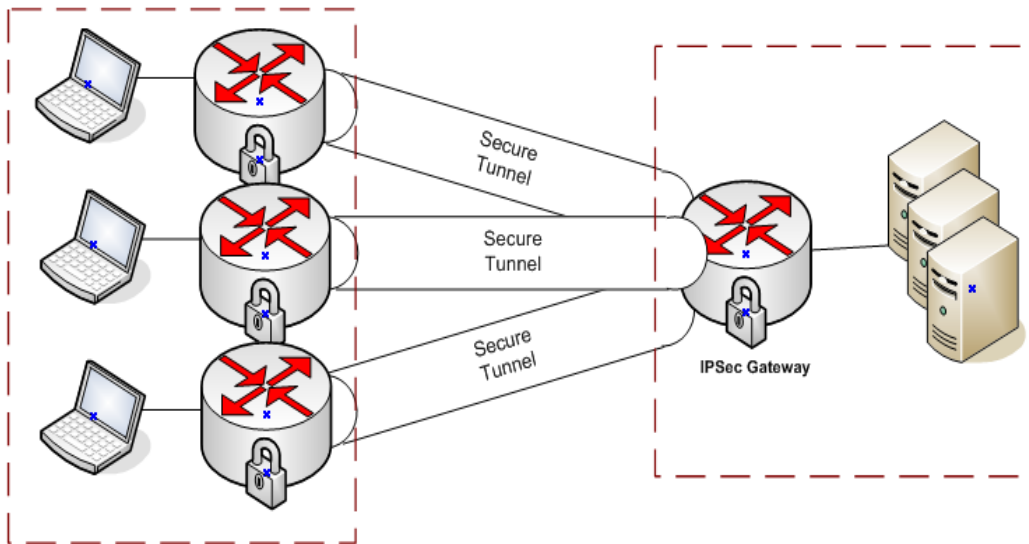


Figure 71. Test Setup

Step-by-Step Instructions

Defining the Network and Traffic Flows

1. Create two Networks (Network1 and Network2).
2. Add the HTTP Client activity to Network1.
3. Add an HTTP Server activity to Network2.

TEST CASE: IPSEC – TUNNEL CAPACITY TEST

- Using the 'lollipop' connector exposed on the right side of the HTTPClient1 activity, drag a symbolic link over HTTPServer1. This will create a basic HTTP configuration for back-2-back. The result is as follows:

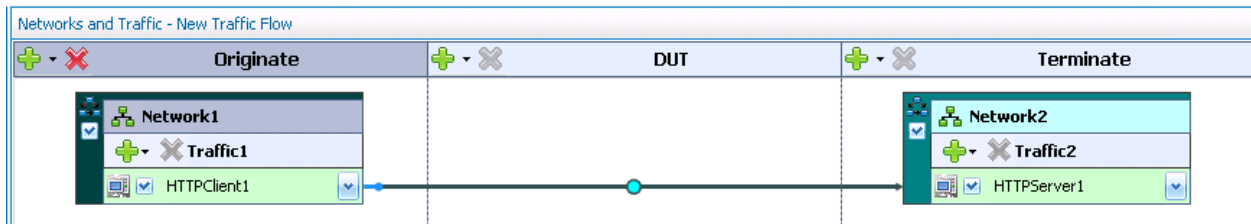


Figure 72. Overview of Network and Traffic Flow

To achieve better data rates, select the activity **HTTPClient1** and change the HTTP page size to /1024k.html (see the parameters of GET 1 command listed under the **Commands** page).

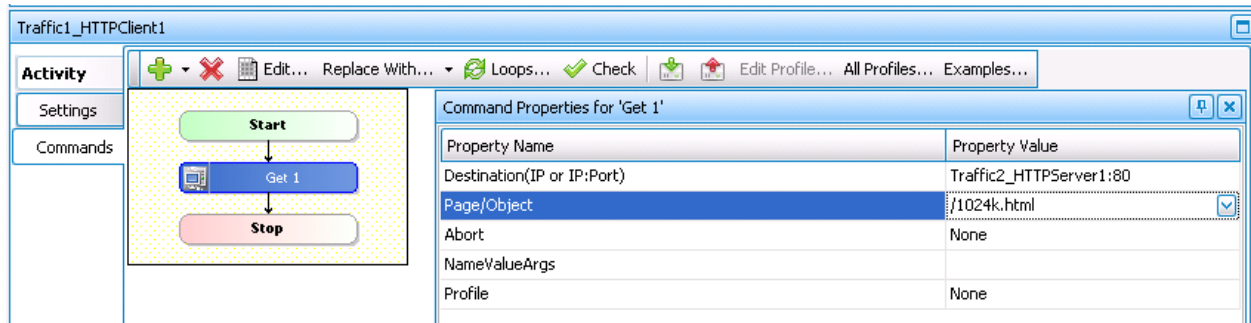


Figure 73. Command Editor - preview of the HTTP GET settings

By default, the throughput objective will seek for the optimal number of users and IPs to generate the maximum data rates. This may result, however, in a small number of IPs generating traffic. Therefore, you can enforce all the IPs to generate data traffic by setting Cycle users through all source IPs for the HTTPClient1 activity. To do so, click **Network 1 | Traffic1**, select the **IP Mappings** page, and then replace Use Consecutive IPs mapping rule with Cycle users through all source IPs by selecting from the list.

Because of memory availability, the maximum number of simulated users is maintained below 10,000 when the number of concurrent tunnels connected is close to the maximum limits of the port.

TEST CASE: IPSEC – TUNNEL CAPACITY TEST

To have traffic from every tunnel, we use Cycle users through all source IPs option as shown in the following figure and we apply a simulated user constraint to the test objective.

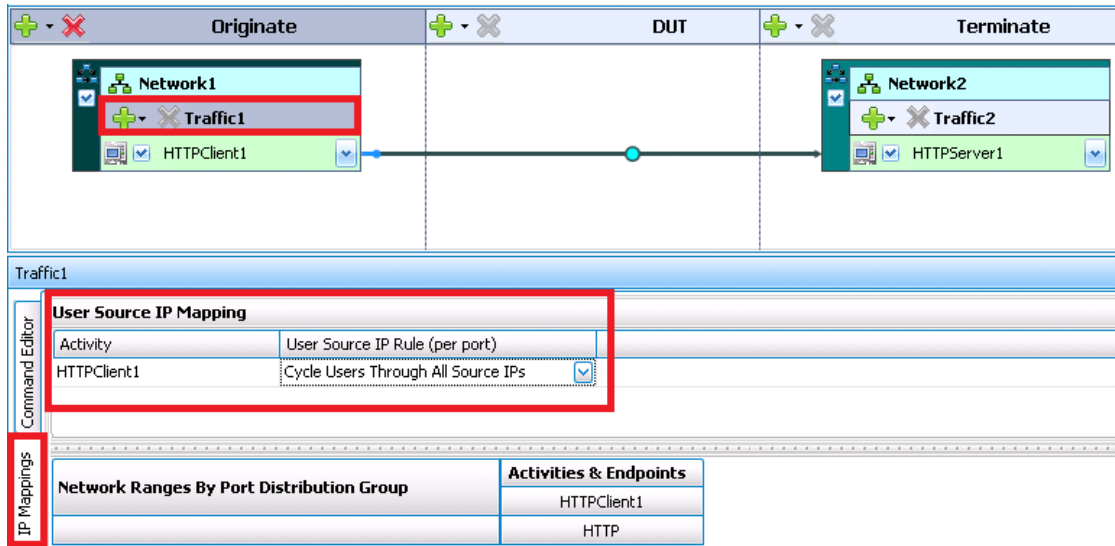


Figure 74. User source IP Mapping

The constraint of simulated users configurable—available as an option for Throughput objective—allows specified number of simulated users to be maintained active. When the number of users applied as a constraint is smaller than the number of IPs defined at the network level, the 'Cycle Users Through All Source IPs' forces the emulated users to cycle through all IPs, allowing every IPsec tunnel to generate traffic at a given time during test execution.

Configuring IPsec settings by using the IPsec Network Wizard

To simplify the IPsec configuration, IxLoad provides an IPsec wizard. To start **IPsec Network wizard**, select the **Wizards menu**, and then click **New IPsec Network Wizard**.

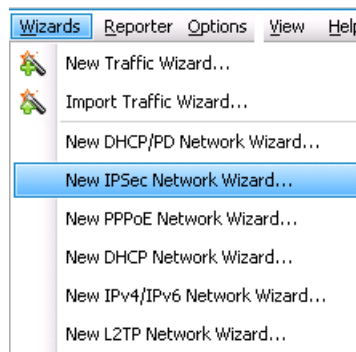


Figure 75. Launch the IPsec Network Wizard

TEST CASE: IPSEC – TUNNEL CAPACITY TEST

1. At the first configuration step of the wizard, set the following:

Test Type: Port-to-Port

Test Scenario: Site-to-Site

IKE Version: IKEv1

Number of IPsec tunnels per Range = 30,000

Unique MAC per EG¹⁰: Checked

Note: This option assigns a unique MAC for each emulated gateway.

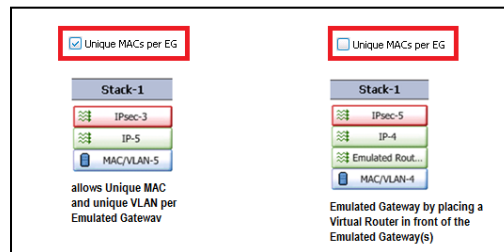


Figure 76. Comparison of the results of Unique MACs per EG

Network1 as IPsec Initiator

Network2 as IPsec Responder

Number of IP/IPsec Ranges per Network Group = 1

¹⁰ Selecting 'Unique MAC per EG' will position the IP network stack directly over the MAC network stack. Clearing the Unique MAC per EG option will result in an intermediate Emulated Router network stack added between the IP and MAC network stacks. When Emulated Router stack is added, all IPsec Emulated Gateways are placed behind a Virtual Router, which allows a single MAC address and VLAN tag to be configured. Removing the MAC layer allows configuration of unique MAC and VLAN per Emulated Gateway.

TEST CASE: IPSEC – TUNNEL CAPACITY TEST

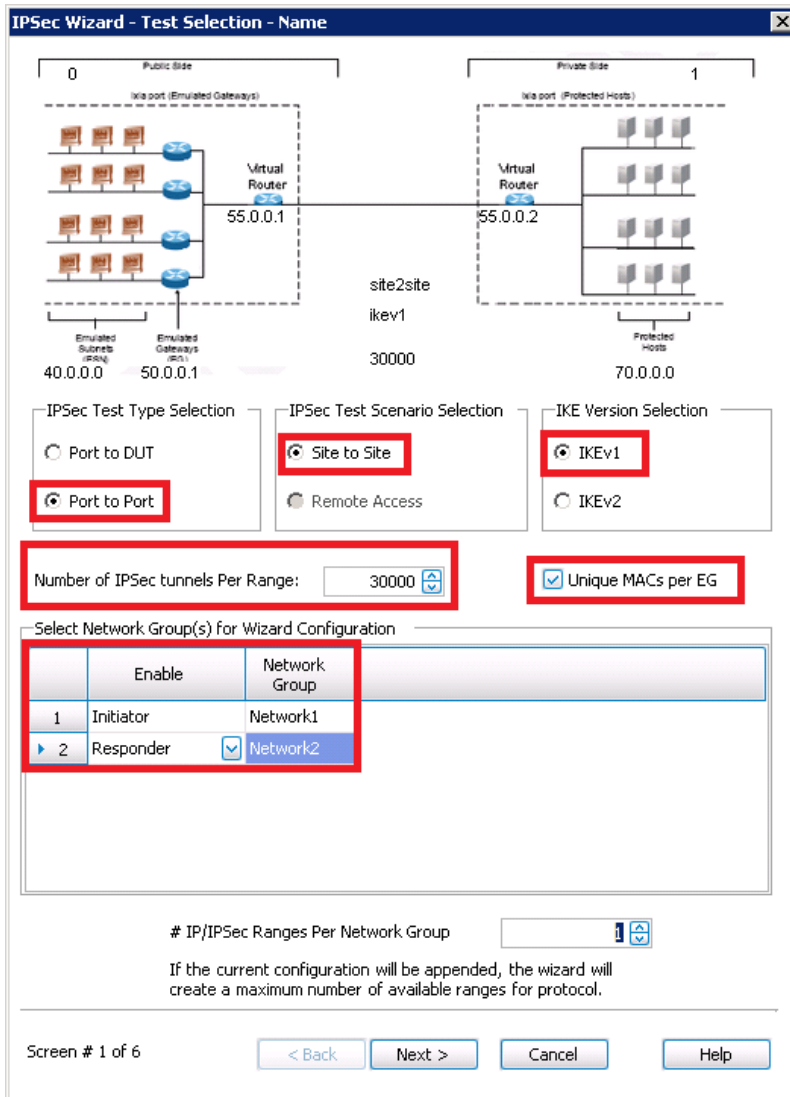


Figure 77. IPsec Network Wizard - screen #1 of 6

TEST CASE: IPSEC – TUNNEL CAPACITY TEST

2. At the second configuration step of the wizard, set the following:

Phase1 Hash Algorithm: HMAC-MD5

Phase1 Encryption: AES-128

Phase2 Hash Algorithm: HMAC-MD5

Phase2 Encryption: AES-128

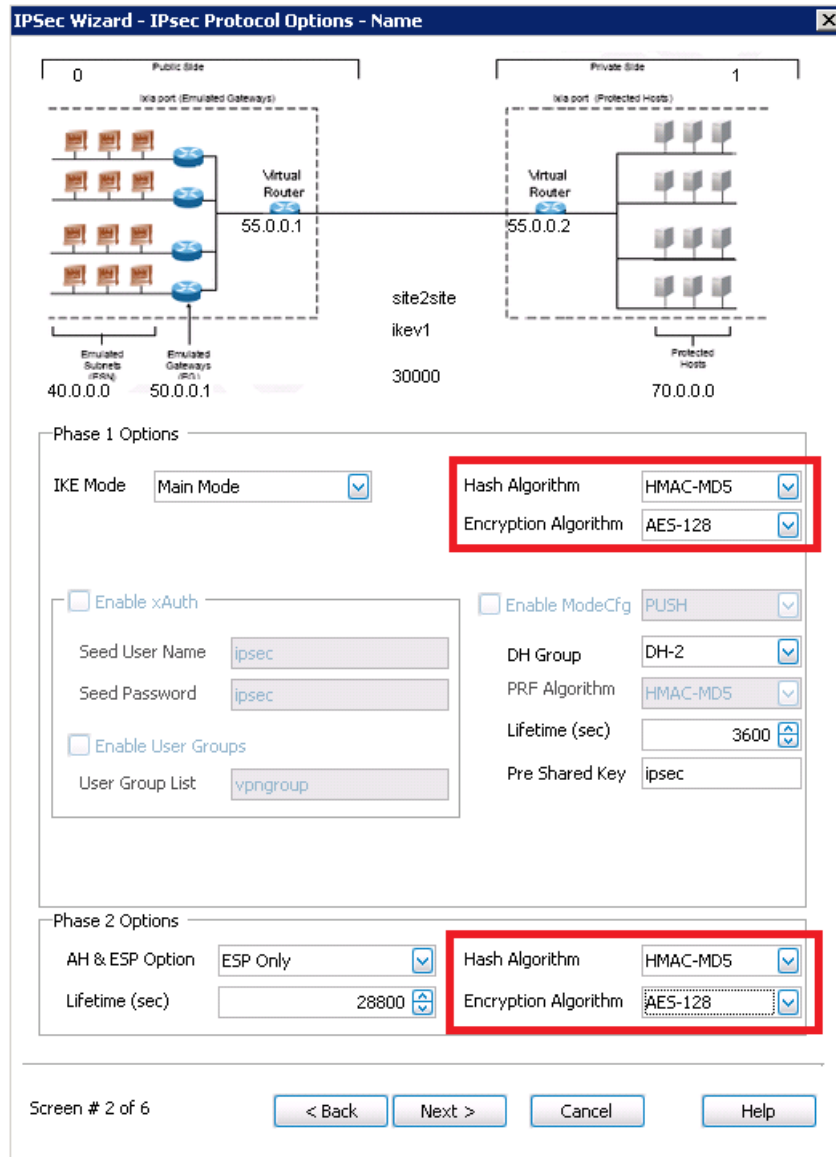


Figure 78. Phase1 and Phase2 Settings

TEST CASE: IPSEC – TUNNEL CAPACITY TEST

3. Leave the other options to their default values:

IKE Mode = Main Mode

AH&ESP = ESP

DH Group = DH-2

PreShared Key = IPsec

Phase1 Lifetime 3600 (sec)

Phase2 Lifetime 28800 (sec)

4. Click **Next** to continue.

TEST CASE: IPSEC – TUNNEL CAPACITY TEST

5. In the wizard's screen number 3:

Set the **Subnet IP Address Type** to IPv6.

Leave the **Gateway IP Address Type** to IPv4.

For the **Emulated Gateways**, set the **Prefix** to 16 (IP mask 255.255.0.0).

The screenshot displays the 'IPsec Wizard - IPsec Network Options - Name' configuration window. At the top, a network diagram shows two sides: 'Public Side' (labeled '0') and 'Private Side' (labeled '1'). The Public Side features a 'Virtual Router' with IP '55.0.0.1' connected to 'Emulated Subnets (50.0.0.1)' and 'Emulated Gateways (50.0.0.1)'. The Private Side features a 'Virtual Router' with IP '55.0.0.2' connected to 'Protected Hosts (4600:0)'. A 'site2site ikev1' tunnel is shown between the routers with a capacity of '30000'. Below the diagram, the 'Subnets IP Address Type' is set to 'IPv6' (highlighted with a red box), and the 'Gateways IP Address Type' is set to 'IPv4'. The 'Public Area Configuration' section includes 'Emulated Gateways' with 'First IP Address' '50.0.0.1', 'Increment By' '0.0.0.1', 'Range Increment Step' '0.0.1.0', and 'Prefix' '16' (highlighted with a red box). The 'Private Area Configuration' section includes 'Emulated Gateways' with 'First IP Address' '60.0.0.1', 'Increment By' '0.0.0.1', 'Range Increment Step' '0.0.1.0', and 'Prefix' '16'. Below this, 'Emulated Subnets' and 'Protected Subnets' are configured with 'First IP Address' '::2800:0' and '::4600:0' respectively, 'Increment By' ':::100', 'Range Increment Step' ':::1:0', 'Num Hosts' '1', and 'Prefix' '120'. A 'DUT Configuration' section at the bottom has empty fields for 'Public IP Address' and 'Private IP Address'. At the very bottom, it says 'Screen # 3 of 6' and has buttons for '< Back', 'Next >', 'Cancel', and 'Help'.

Figure 79. IPsec Network Wizard, configuration screen 3

Note: Starting with the IxLoad 5.00 release, all combinations IPv4/IPv6 are supported, which this includes IPv4/IPv4, IPv6/IPv6, IPv4/IPv6, and IPv6/IPv4,

6. Click **Next** to continue.

TEST CASE: IPSEC – TUNNEL CAPACITY TEST

- At screen number 4 and screen number 5 of the **IPsec Network** wizard, keep all settings to their default values.

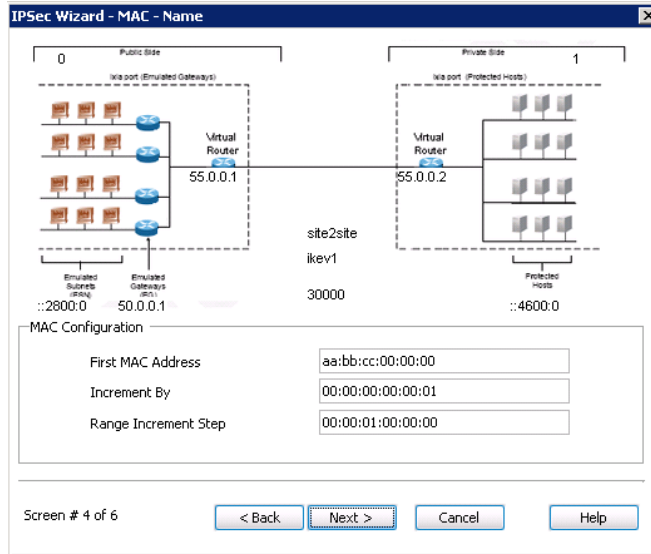


Figure 80. MAC configuration

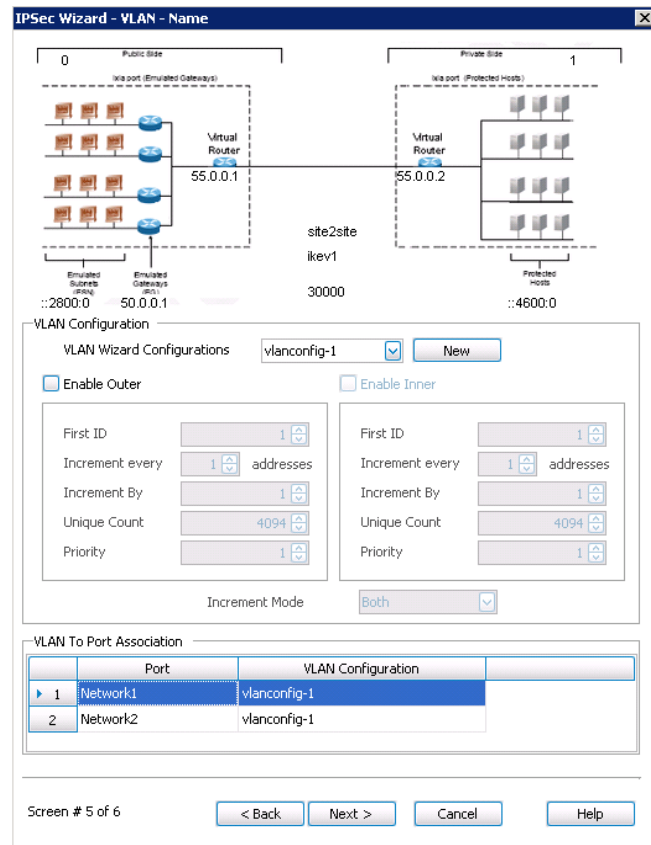


Figure 81. VLAN configuration

TEST CASE: IPSEC – TUNNEL CAPACITY TEST

- Click **Generate and Overwrite Existing Configuration**, and then click **Finish** to apply your configuration.

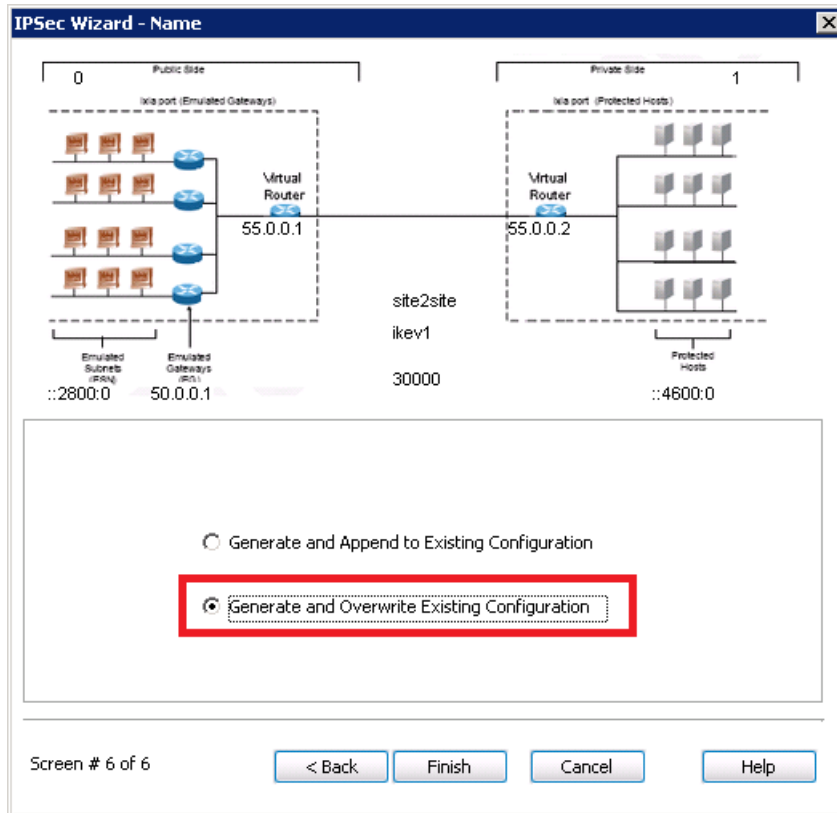



Figure 82. Append or Over write the network configuration

Review and validate the configuration generated by the IPsec Network Wizard

Take a moment to review and understand how the **IPsec Network** wizard configured the parameters of IP and IPsec network stacks for both Network1 and Network2 elements.

Configuring the tunnel setup and tunnel teardown rates

- Select either **Network1** or **Network2**.
- Click **Network Plug-in Global Settings** .

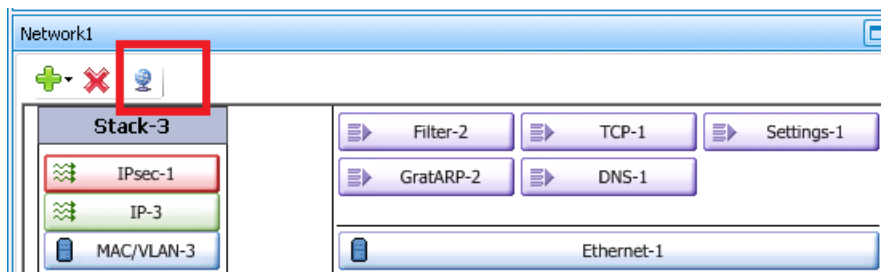


Figure 83. Network Global Settings

- The **Network Plug-In Settings** window is displayed.

TEST CASE: IPSEC – TUNNEL CAPACITY TEST

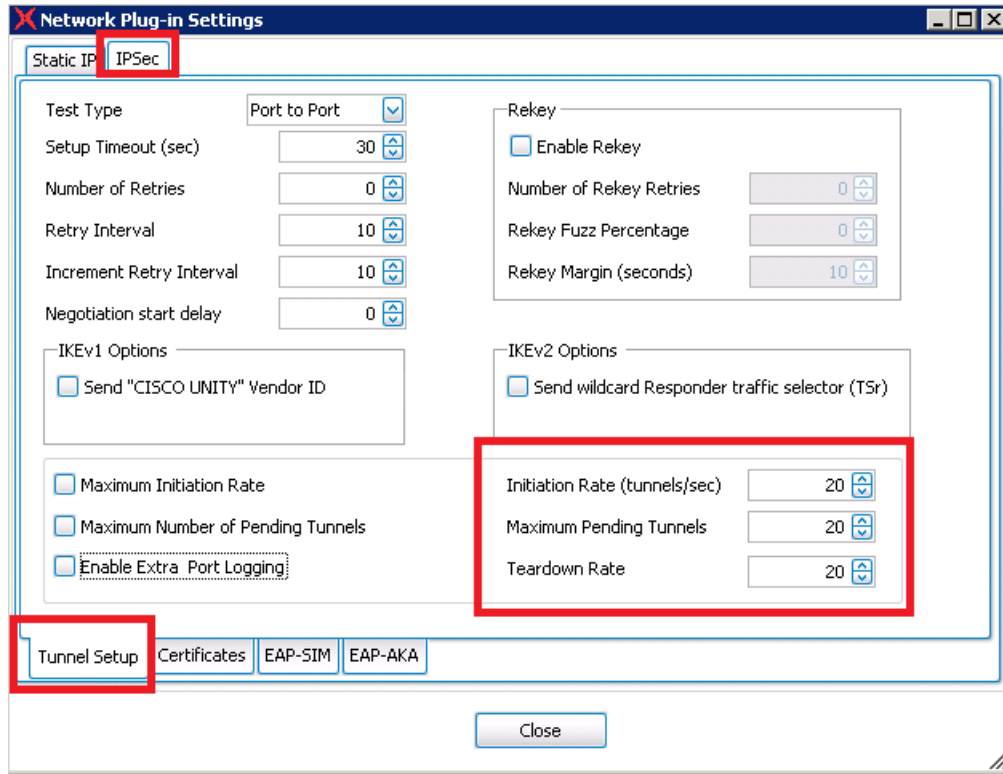


Figure 84. IPsec Global Settings

4. Select the **IPsec** page | **Tunnel Setup** page.
5. Change the following global **parameters**:
 - Set **Initiation Rate** = 20.
 - Set **Maximum Pending Tunnels** = 20.
 - Set **Maximum Teardown Tunnels** = 20.

Test Objective

1. Select the **Timeline and Objective** step.
2. Select the **HTTPClient1** activity to set its objective.
3. Set the test objective as **Throughput (Kbps)** with a value of 100,000 Kbps.
4. Select the **Timeline** configuration page.
5. Set the **Sustain Time** to 3 minutes.

TEST CASE: IPSEC – TUNNEL CAPACITY TEST

- Set the **Ramp Down Time** to 10 seconds.

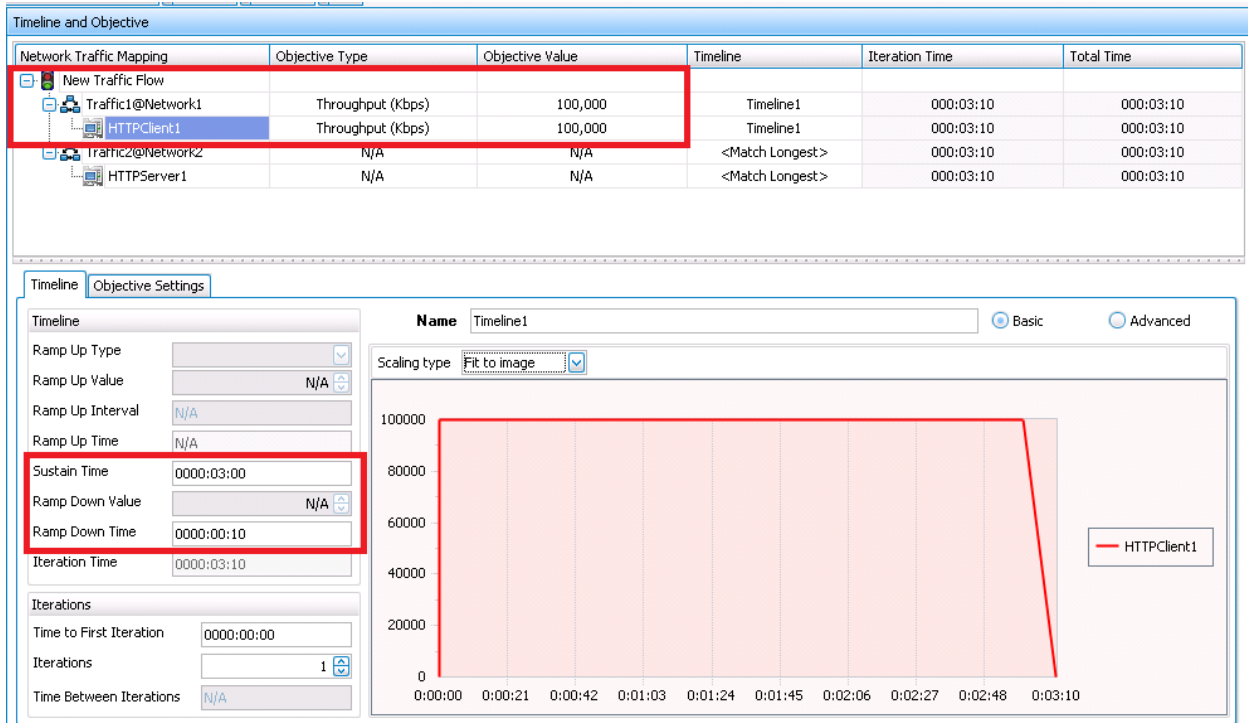


Figure 85. Timeline and Objective configuration

- Select the **Objective Settings** page.
- Set the **Simulated Users** constraint to 25.
- Set the **ramp up value** to 25 users.

Traffic	Objective Type	Objective Value	Constraint En...	Constraint Type	Constraint Value	Ramp Up Value
HTTPClient1	Throughput (Kbps)	12,500	<input checked="" type="checkbox"/>	Simulated Users	25	25

Objective constraints

This setting forces all the IPsec tunnels to generate traffic by cycling users through all IPs using groups of 25 simultaneous users. The 'Cycle users through all source IP' option will have no effect if the simulated user's constraint is set to 100 (matching the number of IP addresses configured).

TEST CASE: IPSEC – TUNNEL CAPACITY TEST

Ports Assignment

The test setup requires two Acceleron NP ports connected back-2-back.

1. From the **Test Configuration** panel, select **Port Assignments**.
2. Add your chassis by clicking **Add Chassis** and by typing the chassis IP.
3. Assign one port to each network.

Test Options

1. Set the **Test Options** and set the following:

Forcefully Take Ownership

Reboot Ports before Configuring

Release Configuration after Test

CSV Polling Interval: 2 [seconds]

Show Diagnostics from Apply Config

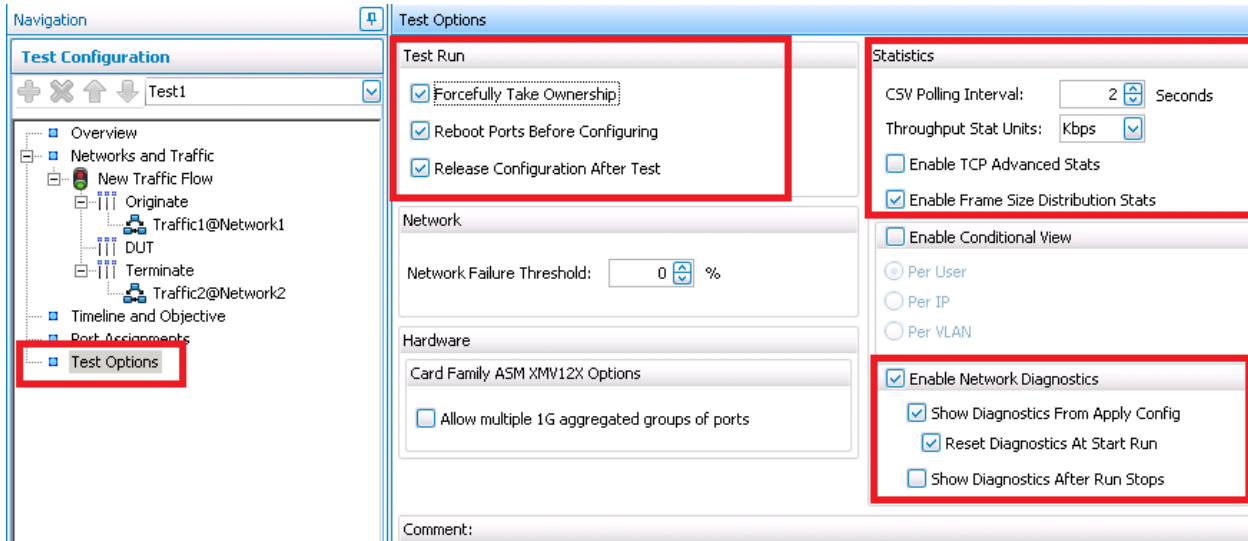


Figure 86. Test options

2. To enable the IPsec stats, make sure you select the **Enable Network Diagnostics** and **Show Diagnostics From Apply Config** check boxes.
3. Save your configuration by clicking **File | Save**.
4. Run the test by clicking **Test | Start** or by clicking **Start** on the toolbar.

TEST CASE: IPSEC – TUNNEL CAPACITY TEST

Test Variables

The main test variables impacting the tunnel capacity are as follows:

Parameter Name	Current Value	Comments
Authentication Method	PreSharedKey	<p>Available options:</p> <p>PreSharedKey, RSA Certificates (one per tunnel or one for all tunnels), EAP (SIM, TLS, AKA, MD5)</p>
Number of L4-7 active users	100	<p>Available options:</p> <p>Less than 10,000 with cycle through when used together with 30,000. tunnels/Acceleron port</p>
DH-Group	DH-2	<p>Available options:</p> <p>DH-1, DH-2, DH-5, DH-14, DH-15, DH-16, DH-18</p> <p>Recommendation: The smaller DH group, the higher the tunnel rate is. Higher DH group numbers offer better security</p>
Traffic Type	Statefull HTTP	Each L4-7 protocol consumes a different amount of memory per user. Experience the results with different application protocols.
Tunnel Flapping Dynamic Control Plane mode	OFF	<p>Available options:</p> <p>On and Off</p> <p>Data Rate Performance</p> <p>Higher degradation when tunnel flapping is enabled as control plane and data plane may share same CPU.</p> <p>Recommended Trials</p> <p>ON</p>

TEST CASE: IPSEC – TUNNEL CAPACITY TEST

Results/Analysis

This section reviews the statistics with main importance for our test:

1. Tunnel Capacity

Review the IPsec statistics indicating the number of Phase2 tunnels initiated, connected, failed, and active by inspecting the following statistics.

Review the number of IPsec sessions initiated.

Review the number of IPsec sessions succeeded.

Review the number of IPsec sessions failed.

Review the number of IPsec active sessions.

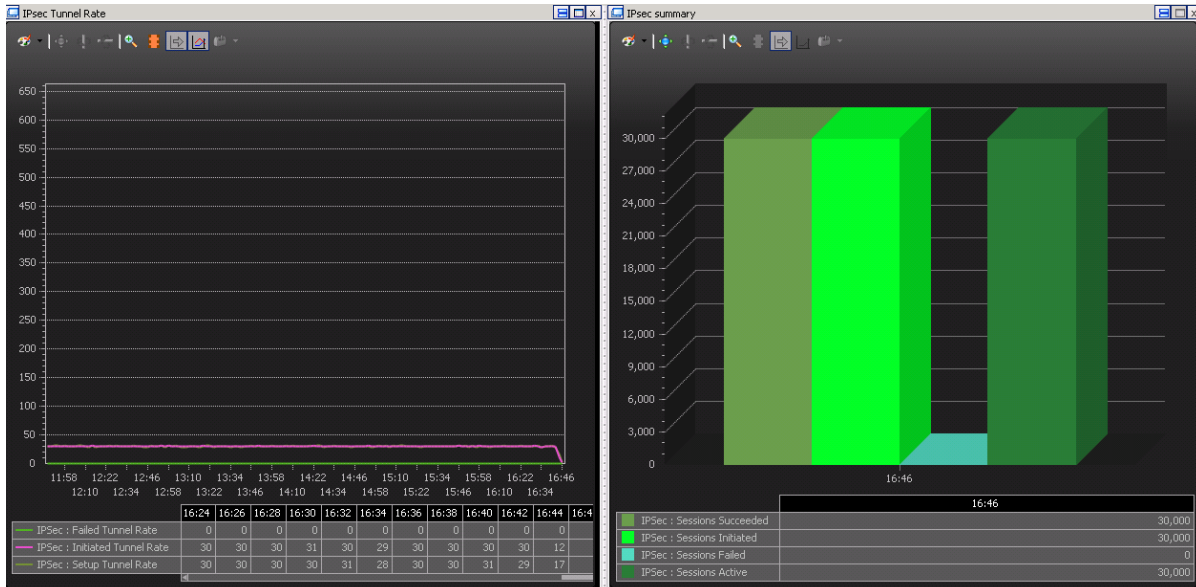


Figure 87. IPsec Tunnel Rates and Tunnel Capacity - sample results

2. Tunnel Rates

Review the IPsec statistics indicating the tunnel initiation rate and the tunnel setup rate, by inspecting the following statistics:

IPsec Tunnel Initiation Rate

IPsec Tunnel Setup Rate

TEST CASE: IPSEC – TUNNEL CAPACITY TEST

L2 versus HTTP throughput

Select L2-3 Throughput statistics view to monitor the L2 TX and RX rate provided individually for both client and server sides.

Select the HTTP – Throughput Objective statistics view to monitor the HTTP Throughput.

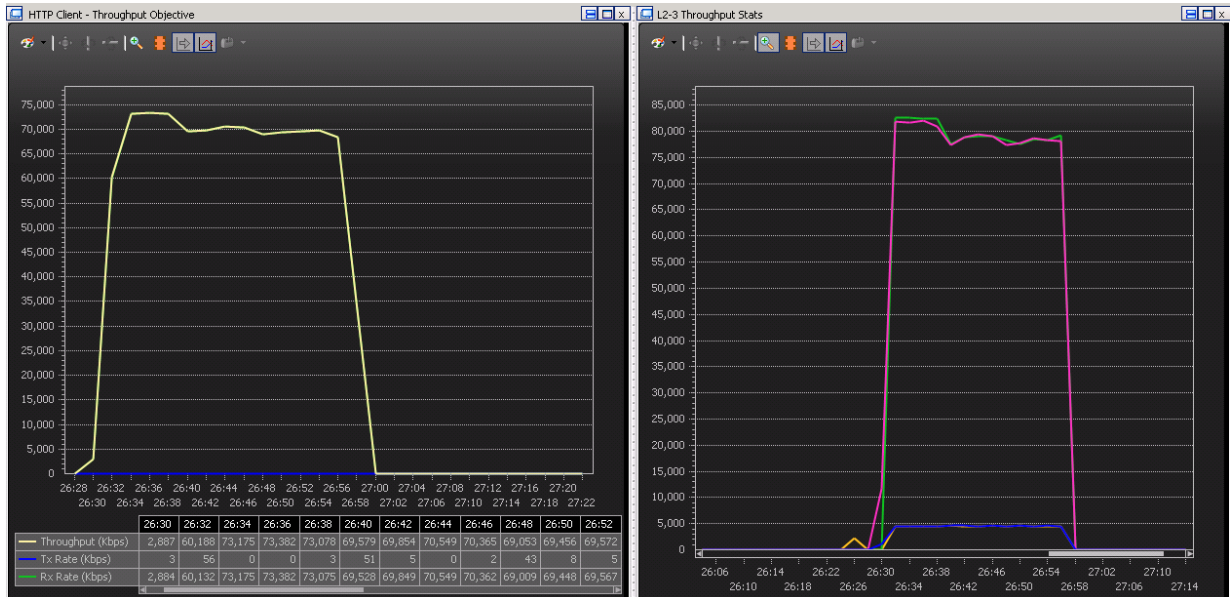


Figure 88. HTTP Throughput vs. L2 Throughput

Conclusions

This test methodology demonstrates how IxLoad can be configured to determine the maximum IPsec tunnel capacity.

Test Case: IPsec Quick Test - RFC 2544 Throughput

Overview

The IPsec Quick Tests are a set of packaged tests designed to benchmark the performance and capacity of IPsec VPN Gateways. You can add, access, run, and customize these tests according to your requirements. These tests are created and stored in the IxLoad configuration file.

IxLoad supports the following IPsec Quick Tests:

- **IPsec Tunnel Setup Rate, Sweep Algorithm** – determines the rate at which IPsec tunnels can be established by a VPN gateway for different Diffie-Hellman groups. This test provides options to include or exclude payload traffic (UDP or HTTP) after the tunnel is established.
- **IPsec RFC2544 Throughput and Latency, Frame loss and Soak** – determines the maximum throughput (encryption rate, decryption rate or bidirectional) at different frame sizes which the DUT can sustain. It also assesses the DUT frame loss behavior for various types of traffic and traffic rates.
- **IPsec Tunnel Capacity** – determines the maximum number of active tunnels that the DUT can sustain concurrently, with or without payload traffic (UDP or HTTP).

The *RFC 2544 IPsec Throughput and Latency* tests measure the throughput and latency of the DUT at different frame sizes. The IPsec throughput is almost identically defined as Throughput in RFC1242], section 3.17. The only difference is that the throughput is measured for a traffic flow that is encrypted, decrypted, or both, by an IPsec device. IPsec throughput is an end-to-end measurement. The configured frame size in the Quick Test is for the payload traffic, and is considered as application throughput also named as goodput.

TEST CASE: IPSEC – QUICK TEST – RFC 2544 THROUGHPUT

Since encryption throughput is not necessarily equal to the decryption throughput, the forwarding rates for both the throughputs must be measured independently.

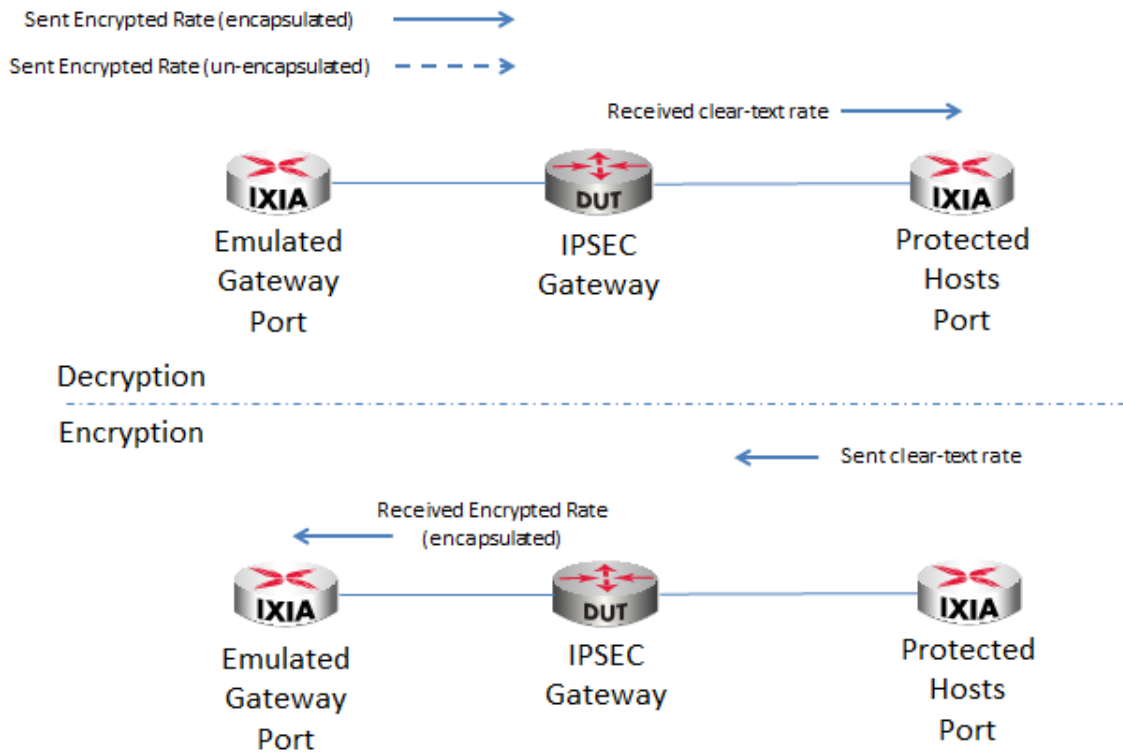


Figure 89. Traffic mapping description

The *Throughput and Latency Quick Tests* use a binary search strategy, in which, the next transmission rate is one half of the difference between the previous successful rate and the previous failed rate. A successful rate is one at which the frame loss is equal to or below the loss tolerance, and a failed rate is one at which it is above the loss tolerance. The binary search continues until the next calculated rate is within configured resolution from the iteration parameters, at which point the test iteration stops. The following figure shows an example of the binary search algorithm using a resolution of 1 Mbps.

In a bidirectional test in which the throughput for a given frame size is different in each direction, the binary search uses the lower of the two throughput values.

TEST CASE: IPSEC – QUICK TEST – RFC 2544 THROUGHPUT

The IPsec encryption and/or decryption rates can be directly impacted by:

- frame size
- the phase 2 encryption and hashing algorithms
- the use of PFS (Perfect Forward Secret) option

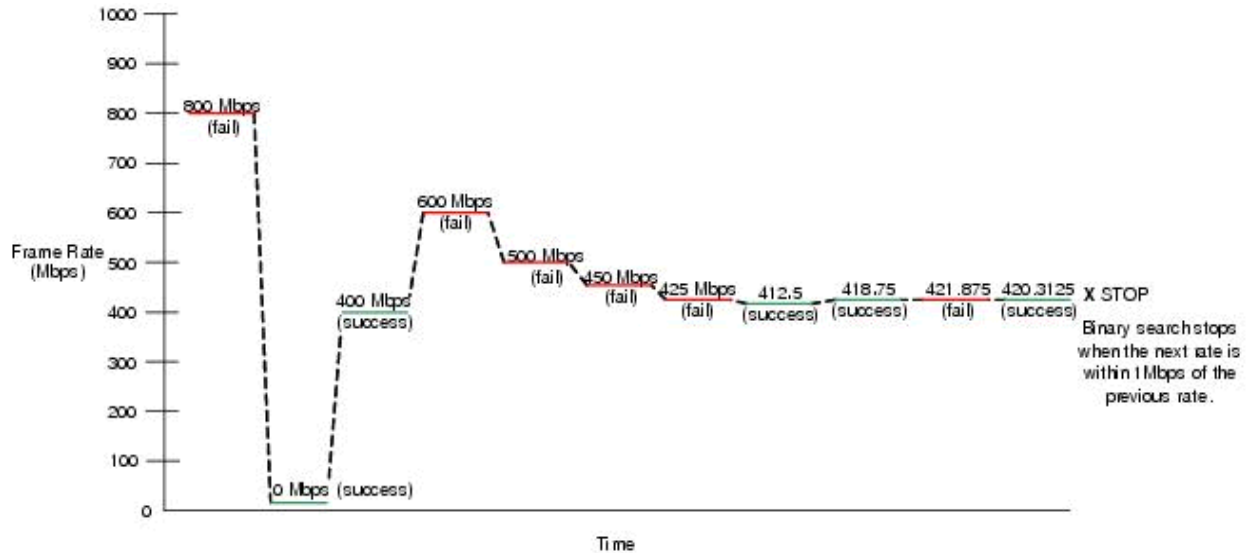


Figure 90. Binary search algorithm conversion example

Issues & Considerations

In certain scenarios, packets are offered to an IPsec Gateway using a frame size that is larger than the MTU of the ingress interface of the IPsec Tunnel transporting the packet. In this case, the packets need to be fragmented before the IPsec services are applied.

In other cases, the packet is of a size very close to the size of the MTU of the egress interface of the IPsec tunnel. In this case, the mere addition of the IPsec header will create enough overhead to make the IPsec packet larger than the MTU of the egress interface. In such instances, the original payload packet must be fragmented either before or after applying the IPsec overhead.

When measuring the IPsec encryption throughput, one has to consider that when probing with packets of a size almost equal to that of the MTUs associated with the IPsec tunnel, fragmentation may occur and the decrypting IPsec device (either a tester or a corresponding IPsec peer) has to reassemble the IPsec and/or payload fragments to validate the original submitted content. If frame loss is detected in case of fragmentation, the algorithm stops after the first iteration and reports that the test step failed.

Objectives

This test case assists you in building the test configuration to use the IPsec RFC 2544 Throughput and Latency Quick Test suite to determine the maximum encryption and decryption

TEST CASE: IPSEC – QUICK TEST – RFC 2544 THROUGHPUT

rates that can be sustained by an IPsec Gateway named from this point on DUT (Device Under Test).

Setup

The test topology consists of a site-to-site deployment where one Ixia test port emulates 100 IPsec gateways connected to the public interface of the DUT (IPsec VPN gateway) and the second test port emulates the protected IP endpoints located behind the private interface of the DUT.

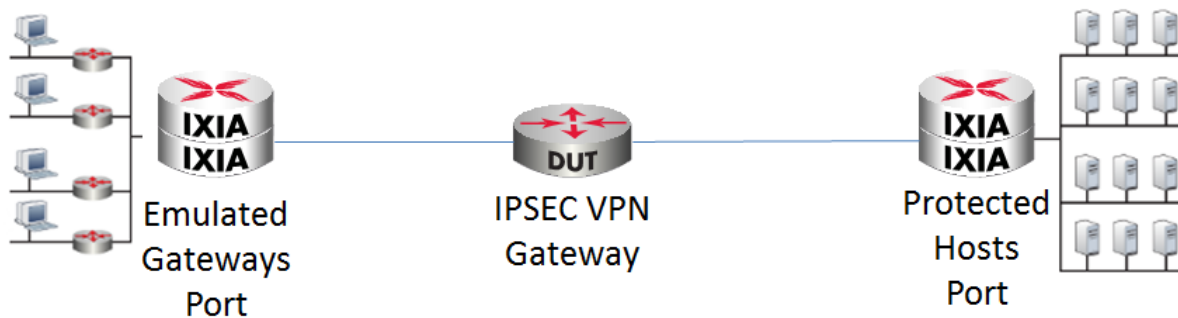


Figure 91. Test Topology

Step-by-Step Instructions

1. Start the IxLoad application.
2. Start the quick test framework by selecting **Quick Tests** entry from the left-pane menu.

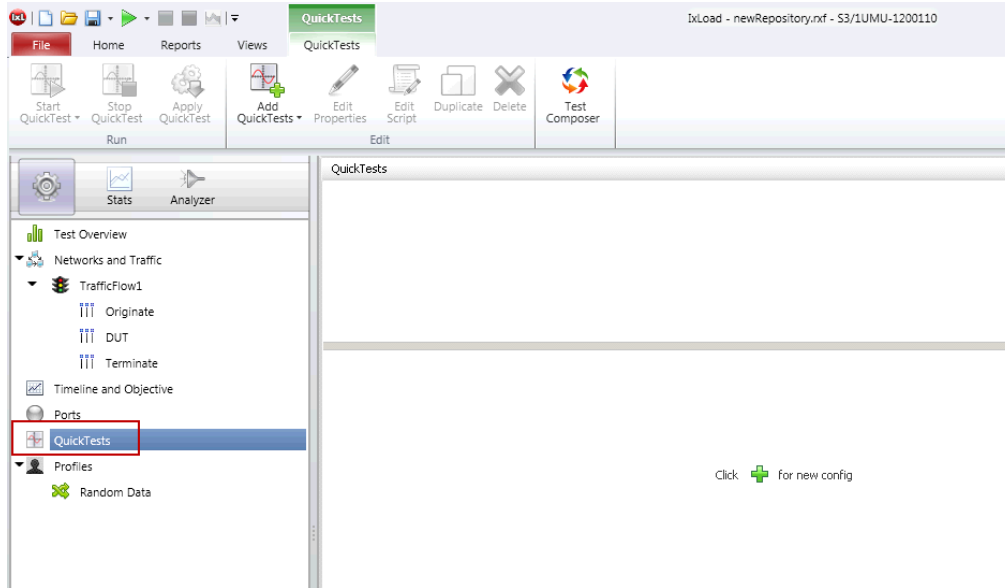


Figure 92. Quick Tests Framework

3. Use the **Add Quick Test** ribbon option to add a new quick test; the Quick Tests wizard starts

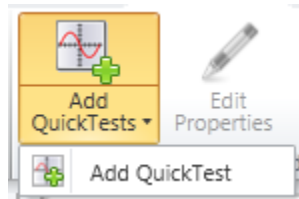


Figure 93. Adding a new test case

TEST CASE: IPSEC – QUICK TEST – RFC 2544 THROUGHPUT

4. Select **IPsec RFC2544 Throughput/Latency** then select **New Configuration** mode.

The **New Configuration** mode builds a configuration from scratch, overriding any previously configured settings.

The **Existing Configuration** mode is designed to build the test methodology over an existing IxLoad configuration.

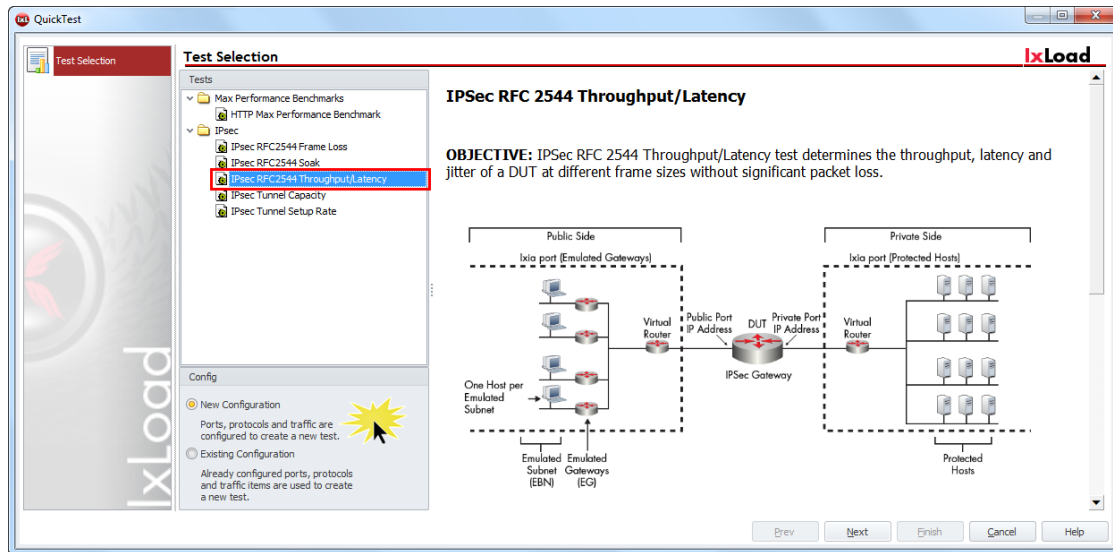


Figure 94. Selecting test scope and configuration mode

5. Select **Port to DUT** test type mode, **Site-to-Site** topology and **IKEv2** as the protocol version. This test case methodology uses IKEv2 with single IP range per Network group for the emulated endpoints.

TEST CASE: IPSEC – QUICK TEST – RFC 2544 THROUGHPUT

6. Select the appropriate Phase 1 and Phase 2 parameters such as hash algorithm, encryption algorithm, DH Group, pre-shared key and key lifetime. Ensure that you configure the same parameters on the DUT to allow a successful tunnel negotiation. Leave the others options to their default values unless changes are necessary.

The image shows two configuration panels for IKEv2. The top panel, 'Phase 1 Options', includes: IKE Mode (Main Mode), Hash Algorithm (HMAC-SHA1), Encryption Algorithm (AES-128-CBC), Enable xAuth (unchecked), Seed User Name (ipsec), Seed Password (ipsec), Enable ModeCfg (unchecked), DH Group (MODP-1024 (2)), PRF Algorithm (HMAC-SHA1), Lifetime (sec) (28800), and Pre Shared Key (ipsec). The bottom panel, 'Phase 2 Options', includes: AH & ESP Option (ESP Only), Hash Algorithm (HMAC-SHA1), Lifetime (sec) (3600), and Encryption Algorithm (AES-128-CBC).

Figure 95. IKEv2 default configuration options

IKE Mode = IKEv2

Hash Algorithm = SHA1

Encryption Algorithm = AES-128

DH Group = DH-2

PreShared Key = ipsec

AH&ESP = ESP Only

Phase1 Lifetime 28800 (sec)

Phase2 Lifetime 3600 (sec)

TEST CASE: IPSEC – QUICK TEST – RFC 2544 THROUGHPUT

7. After setting the security details, configure the network connectivity details according to test environment. The required sets of parameters are:

- IP Version type
- Public subnet IP ranges configuration
- Emulated subnets for traffic generation
- Protected subnets on the private side of DUT
- DUT interface IP details for public and private domain

The screenshot displays a network configuration interface with the following sections:

- Subnets IP Address Type:** IPv4 (dropdown)
- Gateways IP Address Type:** IPv4 (dropdown)
- Public Area Configuration:**
 - Emulated Gateways:** First IP Address: 30.0.0.1, Increment By: 0.0.0.1, Range Increment Step: 1.0.0.0, Prefix: 8
 - Emulated Subnets:** First IP Address: 50.0.0.0, Increment By: 0.0.1.0, Range Increment Step: 10.0.0.0, Num Hosts: 1, Prefix: 24
- Private Area Configuration:**
 - Emulated Gateways:** First IP Address: 40.0.0.1, Increment By: 0.0.0.1, Range Increment Step: 1.0.0.0, Prefix: 8
 - Protected Subnets:** First IP Address: 140.0.0.0, Increment By: 0.0.1.0, Range Increment Step: 10.0.0.0, Num Hosts: 1, Prefix: 24, Single Protected Subnet
- DUT Configuration:** Public IP Address: 22.20.0.1, Private IP Address: 111.222.0.1

Figure 96. Network Configuration details

8. Configure the **Emulated Router** networking details. This represents a “virtual router” emulated on an Ixia port as a network entity routing all the traffic for all the emulated IP addresses. It should be used as a next hop Gateway for the addresses on the public and private side in the DUT routing rules.

Note: Appendix G contains the configuration details required for StrongSwan (www.strongswan.org) IPsec VPN Gateway to successfully establish IPsec tunnels using the details from this test case.

Emulated Router

Emulated Router Configuration	
Public Emulated Router	Private Emulated Router
IP Type: IPv4	IP Type: IPv4
IP Address: 22.20.0.2	IP Address: 111.222.0.2
Increment By: 0.0.0.1	Increment By: 0.0.0.1
Range Increment Step: 0.0.0.2	Range Increment Step: 0.0.0.2
Prefix: 24	Prefix: 24
MSS: 1460	MSS: 1460

Figure 97. Emulated Router Networking details

9. Configure the MAC addresses if necessary for the Ixia emulated entities.
10. Configure the VLAN profiles for the traffic initiated by the Ixia ports. By default all the traffic is untagged. Leave these options unchanged unless necessary.
11. Data Plane settings determine the direction of traffic. There are 3 available options:
 - a. **Encryption** - when the traffic direction is from the private domain to the emulated peers in a public domain. This determines the encryption performance of the DUT.
 - b. **Decryption** – when the traffic direction is from the public domain to the private domain. This determines the decryption performance of the DUT.

- c. **Bidirectional** represents the situation when traffic is transmitted from both sides at the same rate. When assessing the DUT performance in this mode, the lower capacity direction is reported for both the directions measured.

Data Plane Settings

Traffic Type

UDP

UDP Traffic Settings

Source Port 1024

Destination Port 1024

Test Type Encryption

Figure 98. Traffic Options

12. Select the desired traffic direction and the UDP ports for source and destination. For the goal of this test, keep the default settings.
13. Configure the desired Test Parameters for one trial with 1000 IPsec tunnels for a maximum throughput of 100 Mbits.

Trials = 1

Number of tunnels = 1000

Mode = Custom content

Content size = 64, 128,256,512,1024,1280

Content size represents the size of the UDP payload in bytes.

Iteration duration = 1 minute

Throughput Scale = Mbps

Minimum Throughput = 10 Mbps

Maximum Throughput = 100Mbps

Initialization Throughput = 50 Mbps

Traffic Resolution = 1 Mbps

Tunnel Initiation Rate = 100

Maximum pending tunnels = 50

Tunnel Teardown Rate = 100

TEST CASE: IPSEC – QUICK TEST – RFC 2544 THROUGHPUT

Maximum Latency = 100 ms

Maximum Jitter = 100 ms

Note: When using Xcellon-Ultra XTS appliance ports, ensure that you synchronize the nodes by Network Time Protocol to achieve the maximum precision for the measured latencies.

14. Assign the Ixia ports for traffic emulation. The chassis can be the local domain name or the provisioned IP address. After selecting the desired ports, add or remove them to NetTraffics by clicking the corresponding icon.

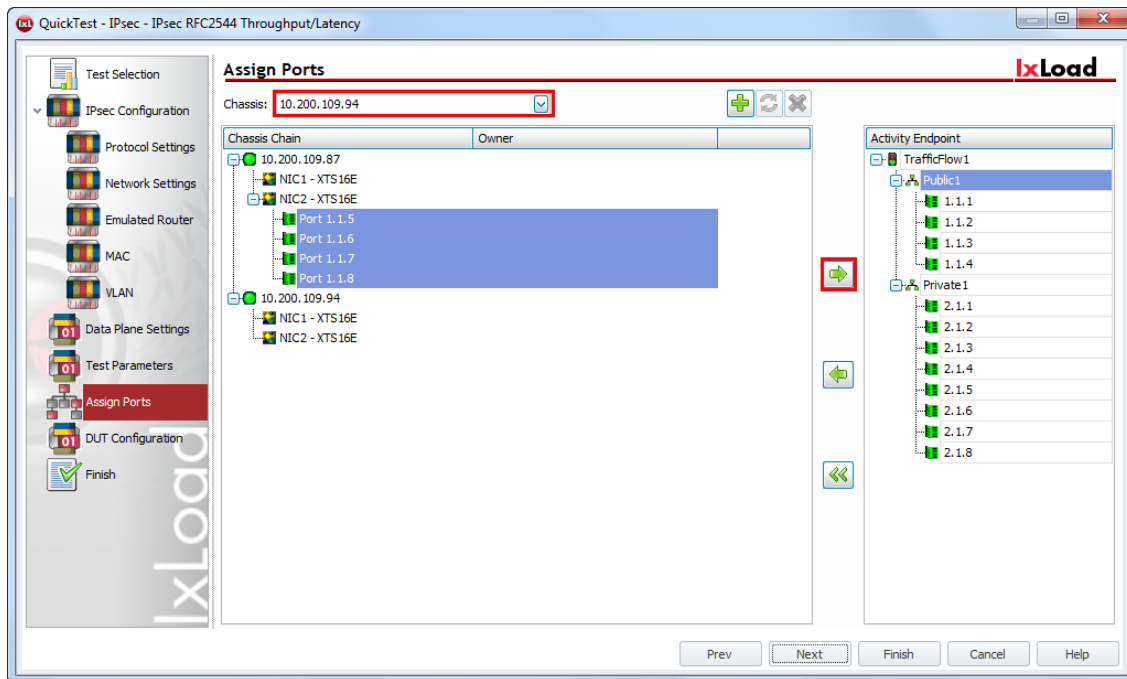


Figure 99. Assigning ports for the emulated networks

TEST CASE: IPSEC – QUICK TEST – RFC 2544 THROUGHPUT

15. Click **Next** until the end of the wizard. As the last step, save the configured Quick Test with a suitable name. You can re-use this configuration later for testing.

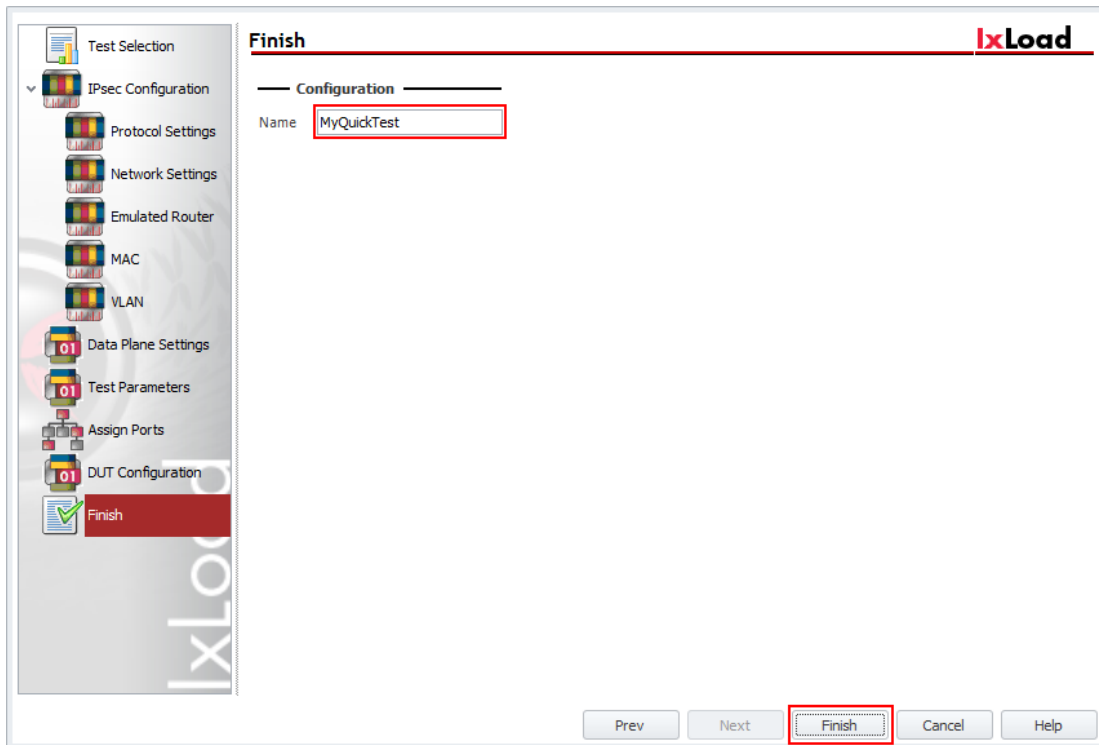


Figure 100. Saving the configured Quick Test

16. Select the desired Quick Test case, click **Start** from ribbon icon to start executing the test.

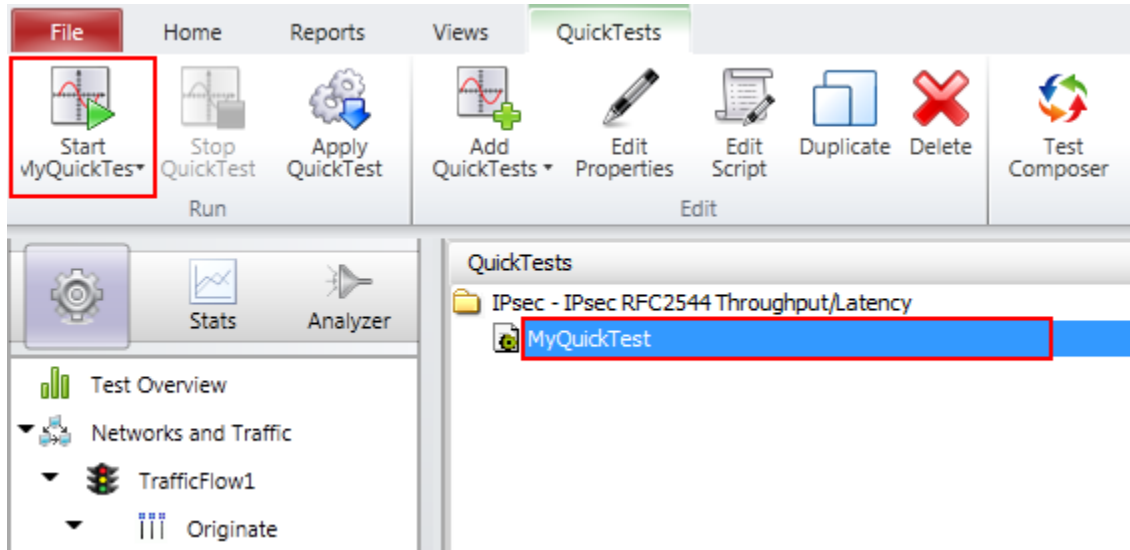


Figure 101. Executing the Quick Test

Test Variables

The important test variables impacting the throughput are as follows:

Parameter Name	Current Value	Comments
Authentication Method	PreSharedKey	Available options: PreSharedKey, RSA Certificates (one per tunnel or one for all tunnels), EAP (SIM, TLS, AKA, MD5)
Encryption Algorithm	AES-128	Available options: DES, 3DES, AES128, AES192, AES256 for both IKEv1 and IKEv2
Content payload size	Custom	Available options: Custom payload size distribution, Range distribution, IMIX representing a distribution of content sizes.
Number of L4-7 active users	100	Available options: Depending on the DUT capabilities these values should be adjusted.
Tunnel Flapping	OFF	Available options:

Parameter Name	Current Value	Comments
Dynamic Control Plane mode		<p>On and Off</p> <p>Data Rate Performance</p> <p>Higher degradation when tunnel flapping is enabled as control plane and data plane may share the same CPU.</p> <p>Recommended Trials</p> <p>ON</p>

Results/Analysis

This section reviews the statistics for the Quick Test:

- Per frame size throughput
- Latency for the measured capacity per frame size

IPsec tunnels

Enable the **IPsec-All ports** statistics view indicating the number of Phase2 tunnels connected to inspect the number of active sessions on the DUT and its performance in handling the traffic.

By enabling the **Total packets** view, you can monitor the number of packets transmitted to the DUT from the public domain and the number of packets received on the private domain by the Ixia emulated peers. The difference might indicate that the maximum throughput performance of the DUT has been reached or exceeded and the binary search algorithm stops reporting the determined value within the configured tolerance.

TEST CASE: IPSEC – QUICK TEST – RFC 2544 THROUGHPUT

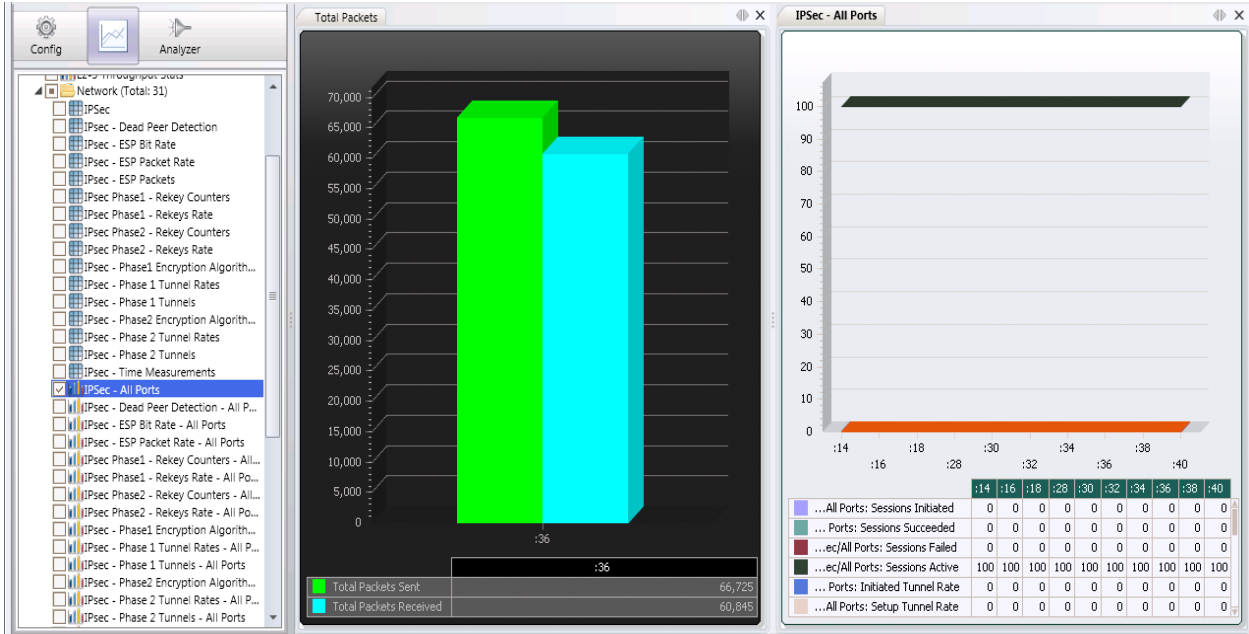


Figure 102. Displaying the total packets statistics and the total IPsec tunnels information

IPsec throughput performance

During the test execution consult the **Throughput** and the **ESP Packet rate** statistics available in the Stat View tree. These provide information about the test measured throughput and indicate the packets per second rate sustained by the DUT for the data payload configured.

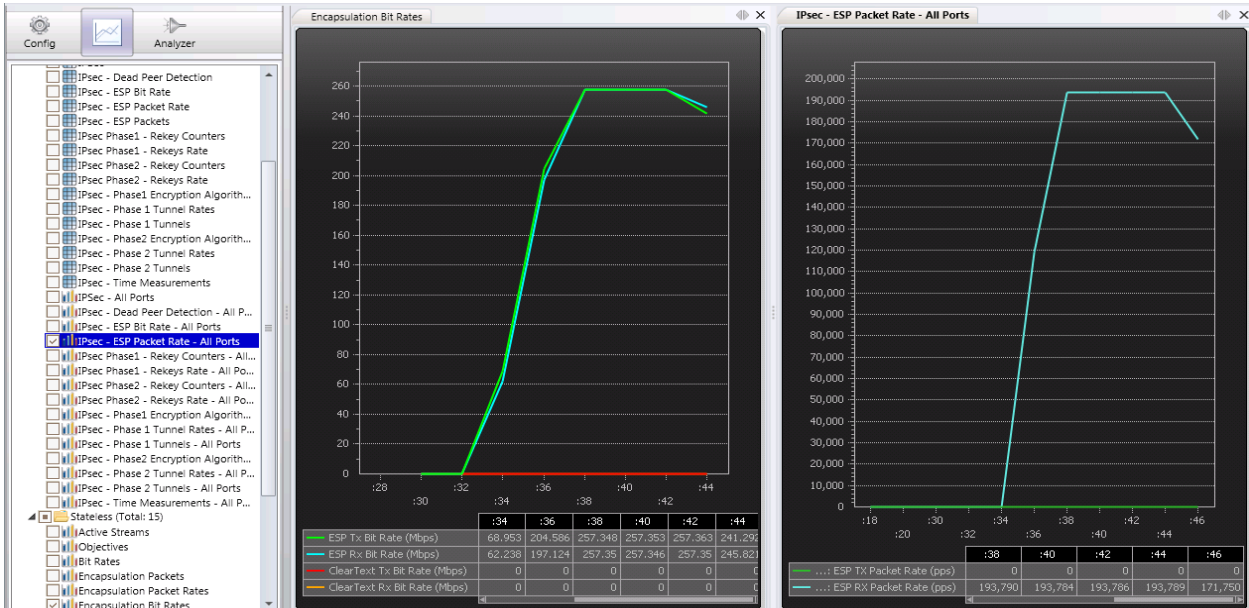


Figure 103. Statistics view for encryption throughput and packet receive rate

You can collect additional information from the Quick Test Log window where the test results of the iterated tests are displayed. The log provides information about the current iteration test for a specific payload throughput and the measured end-to-end latency.

TEST CASE: IPSEC – QUICK TEST – RFC 2544 THROUGHPUT

```
Main Log
11/22/2012 12:06:49 AM: Tunnels Active Decryption Packets Sent Encryption Packets Received Encryption Lost Packet
11/22/2012 12:06:49 AM: *****
11/22/2012 12:06:49 AM: 100 251800 251800 0.000
11/22/2012 12:06:49 AM: *****
11/22/2012 12:06:49 AM: ***** PASS Criteria Evaluation *****
11/22/2012 12:06:49 AM: 0 ms <= 100 ms ?
11/22/2012 12:06:49 AM: *****
11/22/2012 12:06:49 AM: Encryption Latency Criteria : PASSED
11/22/2012 12:06:49 AM: *****
11/22/2012 12:06:49 AM: Pass/Fail Criteria Evaluation Status: PASSED
11/22/2012 12:06:49 AM: *****
11/22/2012 12:06:49 AM: *****
11/22/2012 12:06:49 AM: *****
11/22/2012 12:06:49 AM: ### Encryption throughput ###
11/22/2012 12:06:49 AM: Tx: 93.76 Mbps
11/22/2012 12:06:49 AM: Rx: 93.76 Mbps
11/22/2012 12:06:49 AM: *****
11/22/2012 12:06:49 AM: =====>Binary Search Iteration 5, Framesize 512, Rate 96.875 Mbps, Trial 1, Started 12:06:49
11/22/2012 12:06:49 AM: *****
11/22/2012 12:06:49 AM: Starting IxLoad test
```

Figure 104. Quick Test Log details on test iteration test results

Conclusions

This test methodology assists in configuring an RFC 2544 Throughput and Latency Quick Test suite to measure the Encryption and Decryption performance of an IPsec VPN Gateway using various content payload sizes.

Test Case: IPsec Quick Test – Tunnel Setup Rate

Overview

The IPsec Quick Tests are a set of packaged tests designed to benchmark the performance and capacity of IPsec VPN gateways. You can add, access, run, and customize these tests according to your requirements. These tests are created and stored in the IxLoad configuration file.

The IPsec Tunnel Setup Rate measures the speed at which the VPN Gateway (DUT) can set up increasing numbers of new tunnels with or without data traffic, and how the Setup Rate varies with the Diffie-Hellman group.

IxLoad provides 3 algorithms to determine the tunnel setup rate:

- Sweep
- Binary search and
- Step algorithms

This test case uses the sweep algorithm to determine the tunnel setup rate by measuring the time it takes to establish a group of newly initiated tunnels in parallel. The algorithm uses repeated sweeps to increase the tunnel capacity. The size of the sweep is defined by the user and represents the number of concurrent tunnels that IxLoad initiates at a given moment. A “sweep” starts with the initiation of the tunnels and completes once all the tunnels are successfully established or considered to have failed (no pending tunnels). The next sweep is allowed to start only upon completion of the previous one. Upon completion of each sweep, the algorithm measures its duration and calculates the tunnel setup rate as a ratio between the number of tunnels initiated and the total tunnel setup time (sweep duration). **Example:** A sweep of 100 tunnels takes 500 ms to complete. The calculated setup rate is 200 tunnels/second (100 tunnels/500 ms = 200 tunnels/second)

Depending on the number of IPsec tunnels initiated per sweep and the maximum number of IPsec tunnels configured by user, the Tunnels Setup Rate determination may require one or more iterations. After completing the first sweep, a new set of tunnels is attempted, increasing the overall number of IPsec tunnels that are active on the DUT. The sweeping process continues until the maximum tunnel capacity is reached or until the configured stop criteria is met. The stop criteria for Tunnels Setup Rate can be configured as percentage or total count of tunnels failed.

Objectives

This methodology provides step by step instructions on how to use the sweep algorithm of IxLoad's IPsec Tunnel Setup Rate quick test to quickly determine the maximum tunnel setup rate that can be sustained by an IPsec Gateway.

The following IPsec parameters are the ones that are more likely to impact the tunnel setup rate of the IPsec gateway:

- DH Group (DH-1, DH-2, DH-5, DH-14)
- Authentication method
- the Phase 1 Encryption and hashing algorithms in use

Setup

The test topology consists of a site-to-site deployment where one Ixia test port emulates 100 IPsec Gateways connected to the public interface of the DUT (IPsec VPN gateway) and the second test port emulates the protected IP endpoints located behind the private interface of the DUT.

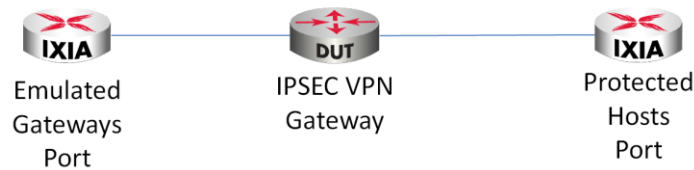


Figure 105. Test Topology

Step-by-Step Instructions

1. Start the IxLoad application.
2. Navigate from left option tree to **Quick Tests** option and select it.

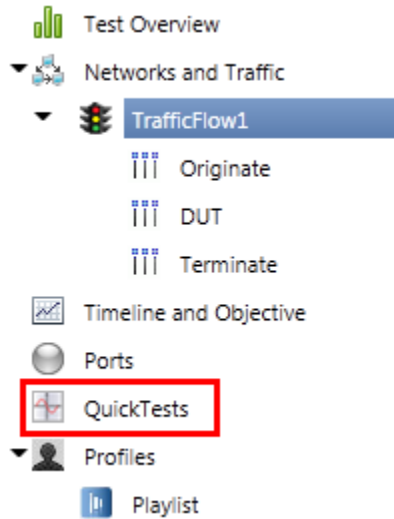


Figure 106. Quick Tests Suite

3. Add a new quick test by triggering the addition from the ribbon icon **Add Quick Test**.

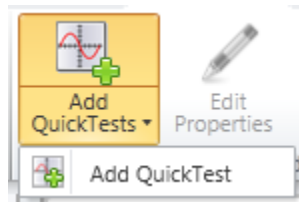


Figure 107. Adding a new test case

TEST CASE: IPSEC QUICK TEST – TUNNEL SETUP RATE

4. In the new window select **IPsec Tunnel Setup Rate** option and **New Configuration** mode. This represents a new scenario with no details configured for setup rate measurement.

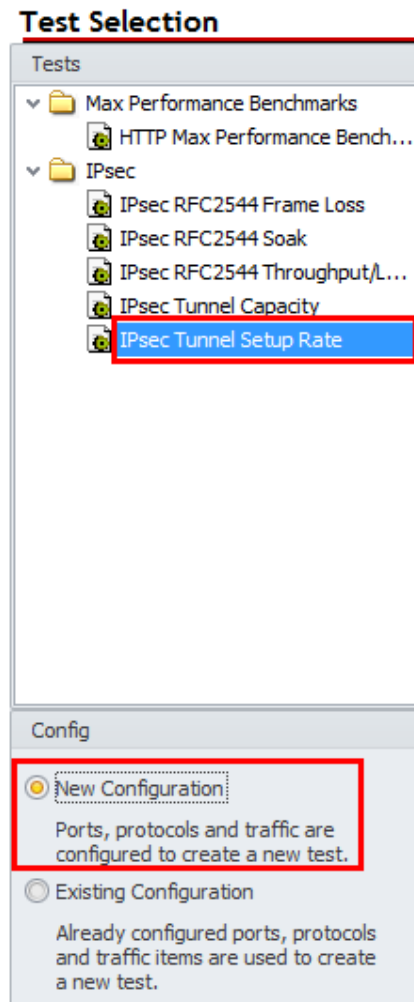


Figure 108. Quick Test suite options

TEST CASE: IPSEC QUICK TEST – TUNNEL SETUP RATE

5. In the **IPsec Configuration** screen select the test type as **Port to DUT** and **Site to Site**. Select **IKEv1** as the selection for IKE version in use. **Enable** the option **Unique MACs per EG** (emulated Gateway). This disables the use of emulated router, allowing each IP address to advertise its own MAC address.

IPsec Configuration

The screenshot shows three selection boxes for IPsec configuration. The first box, 'IPsec Test Type Selection', has 'Port to DUT' selected. The second box, 'IPsec Test Scenario Selection', has 'Site to Site' selected. The third box, 'IKE Version Selection', has 'IKEv1' selected. Below these boxes, a checkbox labeled 'Unique MACs per EG' is checked.

Figure 109. IPsec test scenario details

6. Configure the IKE Mode, encryption and hashing algorithms for Phase 1 and Phase 2, lifetime values and preshared key details. Leave the other options with their default values unless the active DUT configuration requires it. Consult Appendix G for the configuration samples on StrongSwan IPsec VPN Gateway.

Phase 1 IKE Mode = Main Mode

Hashing Algorithm = SHA1

Encryption Algorithm = AES-128

Key Lifetime = 28800 seconds

Preshared Key = ipsec

Traffic Resolution = 1 Mbps

Tunnel Initiation Rate = 100

Maximum pending tunnels = 50

Phase 2 AH& ESP Option = ESP Only

Hashing Algorithm = SHA1

Encryption Algorithm = AES-128

TEST CASE: IPSEC QUICK TEST – TUNNEL SETUP RATE

The image shows two configuration panels for IKEv1. The top panel, 'Phase 1 Options', includes: IKE Mode (Main Mode), Hash Algorithm (HMAC-SHA1), Encryption Algorithm (AES-128-CBC), Enable xAuth (unchecked), Seed User Name (ipsec), Seed Password (ipsec), Enable ModeCfg (unchecked), PRF Algorithm (HMAC-SHA1), Lifetime (sec) (28800), Enable User Groups (unchecked), and User Group List (vpngroup). The bottom panel, 'Phase 2 Options', includes: AH & ESP Option (ESP Only), Hash Algorithm (HMAC-SHA1), Lifetime (sec) (3600), and Encryption Algorithm (AES-128-CBC).

Figure 110. IKEv1 default configuration options

7. Click the **Next** button in the wizard to configure the network specific details. Adjust the required IP addresses to match the test environment. This test case uses the default values but modifications should be done for the setup to allow a successful tunnel connection. Below are the required details to configure the **Network Settings**:
 - IP Version type
 - Public subnet IP ranges configuration
 - Emulated subnets for traffic generation
 - Protected subnets on the private side of DUT
 - DUT interface IP details for public and private domain

Network Settings

Subnets IP Address Type: IPv4 | Gateways IP Address Type: IPv4

Public Area Configuration

Emulated Gateways

First IP Address: 30.0.0.1
 Increment By: 0.0.0.1
 Range Increment Step: 1.0.0.0
 Prefix: 8

Emulated Subnets

First IP Address: 50.0.0.0
 Increment By: 0.0.1.0
 Range Increment Step: 10.0.0.0
 Num Hosts: 1 | Prefix: 24

Private Area Configuration

Emulated Gateways

First IP Address: 40.0.0.1
 Increment By: 0.0.0.1
 Range Increment Step: 1.0.0.0
 Prefix: 8

Protected Subnets

First IP Address: 140.0.0.0
 Increment By: 0.0.1.0
 Range Increment Step: 10.0.0.0
 Num Hosts: 1 | Prefix: 24
 Single Protected Subnet

DUT Configuration

Public IP Address: 22.20.0.1 | Private IP Address: 111.222.0.1

Figure 111. Network configuration settings

- Configure the MAC address settings for unique range and patterns of addresses to be generated for the simulated IP address. Unless necessary, do not change the default MAC ranges.

MAC

MAC Configuration

First MAC Address: aa:bb:cc:00:00:00
 Increment By: 00:00:00:00:00:01
 Range Increment Step: 00:00:01:00:00:00

Figure 112. MAC address range options

- Configure the VLAN profiles for the traffic initiated by the Ixia ports. By default, all the traffic is untagged. Leave these options un-changed unless necessary.
- IxLoad's Quick Test offers the option to generate traffic over the established tunnels. By default the **Data Plane** option is **Disabled**, which means no traffic is transmitted

TEST CASE: IPSEC QUICK TEST – TUNNEL SETUP RATE

performed over the tunnels. There are two options for traffic generation, stateless UDP streams with customizable content size and rate, and HTTP traffic for specific page files. However, the data traffic has an impact on the rate at which the tunnels are set up for the device under test.

This test does not use any traffic over the established tunnels, so the traffic type option should remain disabled. However, this option can be enabled, generating a new test case for the DUT performance measurement.

11. On the test parameters screen adjust the test parameters using the below example. This test scenario will test tree DH groups (DH-1, DH-2 and DH-14) but depending on the DUT configuration additional can be configured.

Trials = 1

Maximum Number of tunnels = 1000

Load type = Sweep

Initial Numer of tunnels = 100

Increment = 50

Acceptable tunnels failures = 100%

DH Groups = DH-1, DH-2 and DH-14

TEST CASE: IPSEC QUICK TEST – TUNNEL SETUP RATE

Note: By configuring the Acceptable tunnels failures to 100%, the sweep algorithm stops measuring the Tunnels Setup rate when the maximum number of tunnels are set up or when no tunnels are established on the iteration step due to DUT's maximum tunnels capacity.

Test Parameters **ixLoad**

Test Parameters

Trials:

Maximum Number Of Tunnels:

Leave tunnels up after test complete

Iteration Parameters

Load Type:

Initial Number Of Tunnels:

Increment:

Algorithm Stop Criteria

Acceptable Tunnel Failures: %

DH Group parameters

- DH Group-01 (MODP768)
- DH Group-02 (MODP1024)
- DH Group-05 (MODP1536)
- DH Group-14 (MODP2048)
- DH Group-15 (MODP3072)
- DH Group-16 (MODP4096)
- DH Group-17 (MODP6144)
- DH Group-18 (MODP8192)

Figure 113. Test options for sweep algorithm

TEST CASE: IPSEC QUICK TEST – TUNNEL SETUP RATE

12. Assign the Ixia ports for traffic emulation. The chassis can be the local domain name or the provisioned IP address. After selecting the desired ports, add or remove them to NetTraffics by clicking the corresponding icon.

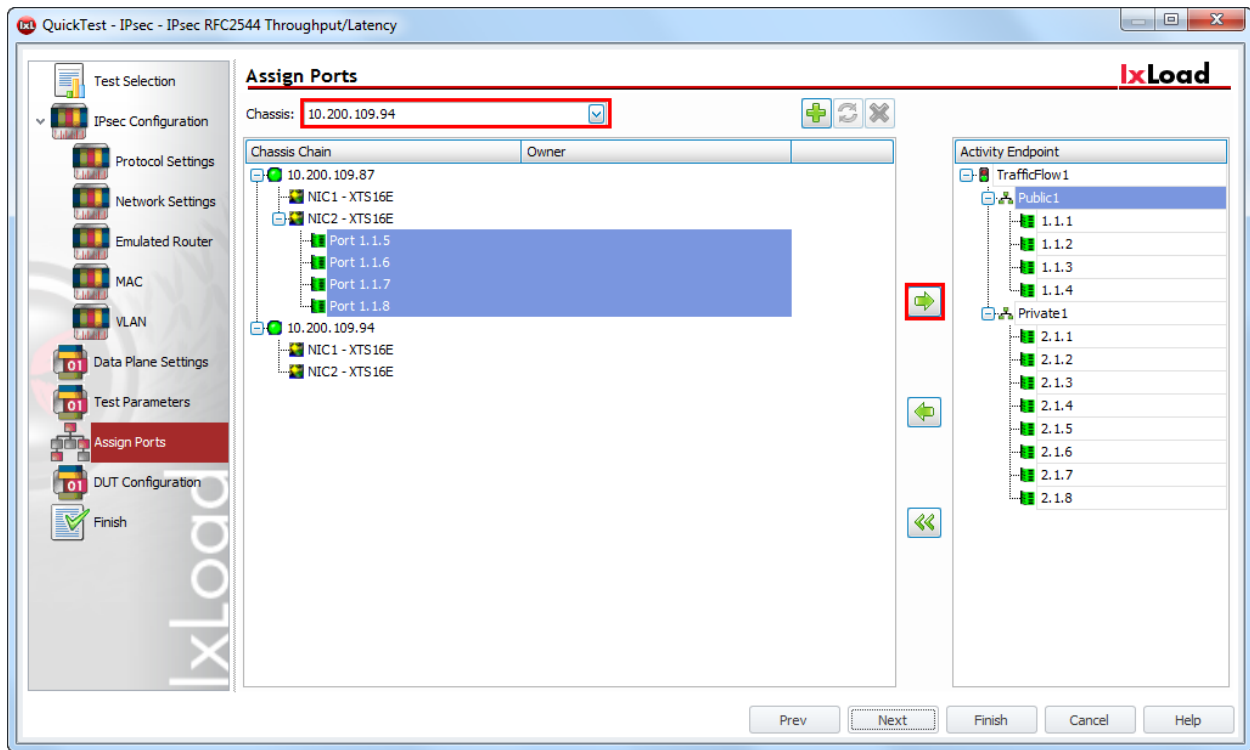


Figure 114. Assigning ports for the emulated networks

13. Click **Next** until the end of the wizard. As the last step, save the the configured Quick Test with a suitable name. Optionally, save the IxLoad configuration file (RXF) for future use.

TEST CASE: IPSEC QUICK TEST – TUNNEL SETUP RATE

14. Select the desired Quick Test case; click **Start** from ribbon icon to start executing the test.

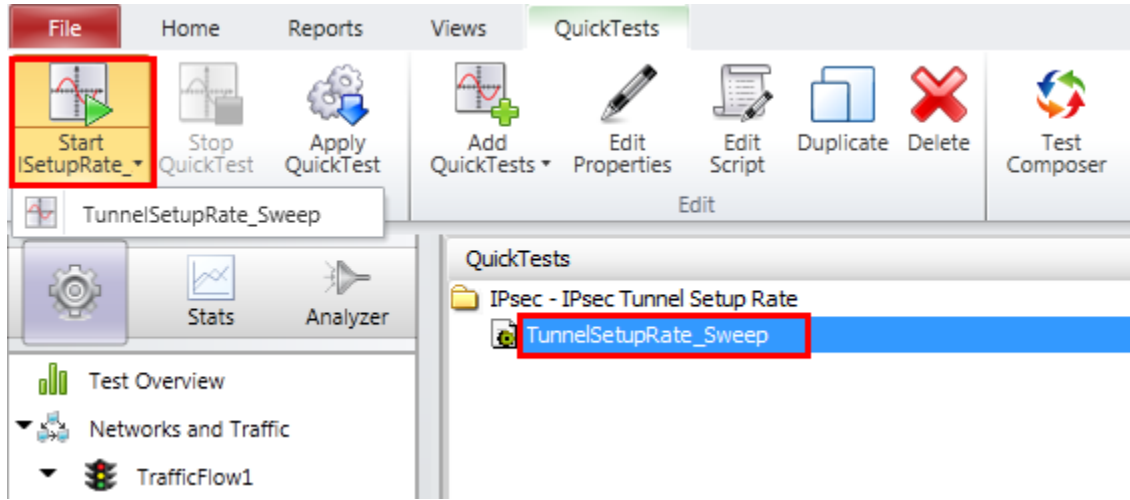


Figure 115. Starting the Quick Test execution

Test Variables

The main test variables impacting the Tunnel Setup Rate are as follows:

Parameter Name	Current Value	Comments
Diffie-Hellman Group	DH-1, DH-2, DH-14	Available options: DH-1(MODP-768), DH-2 (MODP-1024), DH-5 (MODP-1536), DH-14 (MODP-2048), DH-15 (MODP-3072), DH-16 (MODP-4096), DH-17 (MODP-6144), DH-18 (MODP-8192), DH-22 (MODP-1024-5160), DH-23 (MODP-2048-5224), DH-24 (MODP-2048-5256), DH-25 (ECP-192), DH-26 (ECP-224), DH-19 (ECP-256), DH-20 (ECP-384), DH-21 (ECP-521)
Authentication Method	PreSharedKey	Available options: PreSharedKey, RSA Certificates with different key lengths (one per tunnel or one for all tunnels), EAP (SIM, TLS, AKA, MD5)
Encryption Algorithm	AES-128	Available options: DES, 3DES, AES128, AES192, AES256 for both IKEv1 and IKEv2

TEST CASE: IPSEC QUICK TEST – TUNNEL SETUP RATE

Parameter Name	Current Value	Comments
Data plane traffic type	Disabled	Available options: UDP Stateless traffic or HTTP 1.1

Results/Analysis

This section reviews the statistics for the Quick Test Tunnel Setup Rate:

- Average Tunnel Setup Rate
- Minimum and Maximum Tunnel Setup Time

When performing the test, access the **Quick Test Log** window for runtime status of the test results. The current iteration state and the achieved test results for executed trials are displayed in this log. To view the global test status and eventual warning and errors, access the **Main Log** window, available by default at the bottom of the application window.

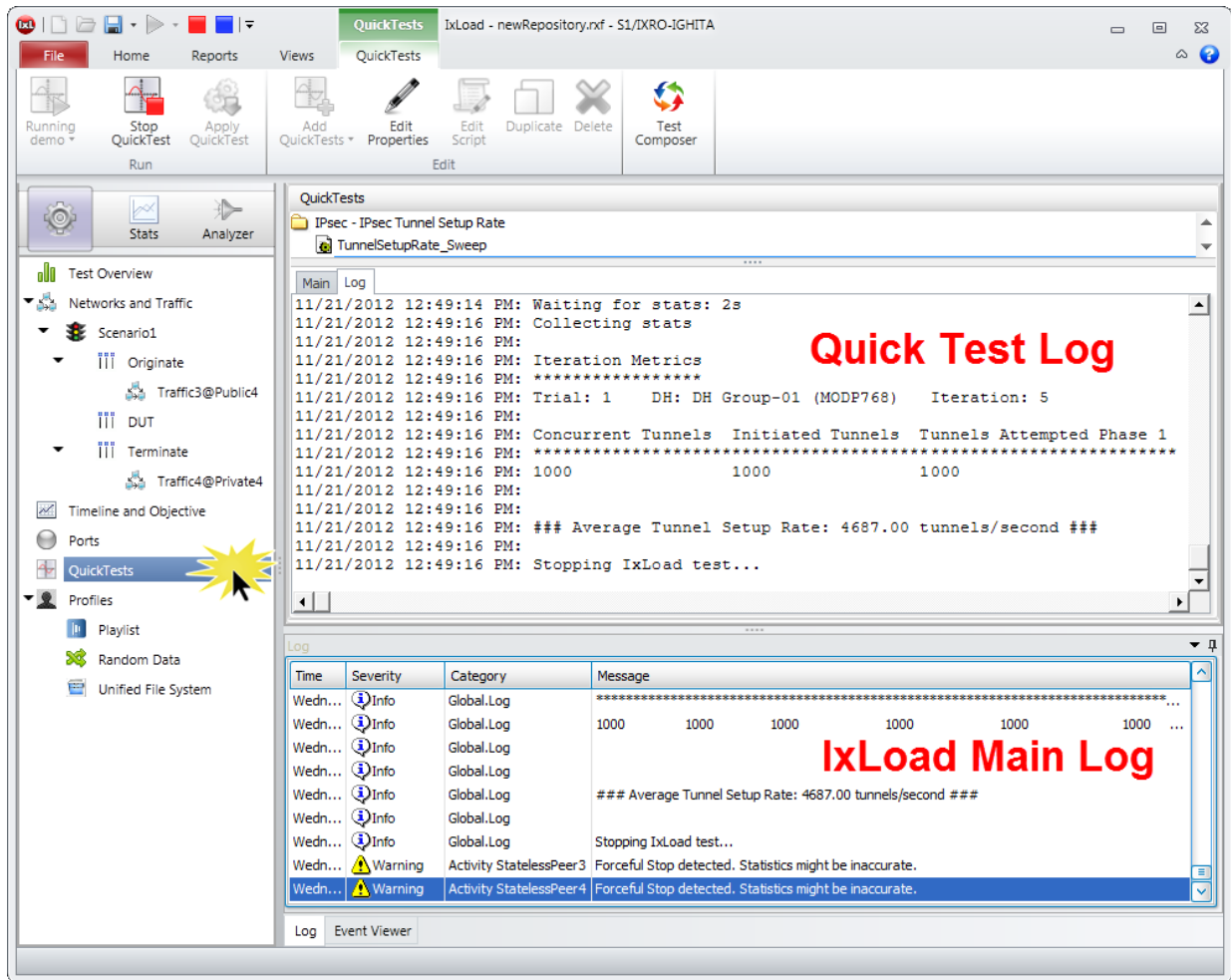


Figure 116. Application layout for logging information

TEST CASE: IPSEC QUICK TEST – TUNNEL SETUP RATE

To monitor test results consult the available statistics from IPsec Tunnels view and IPsec Tunnel Rates view. These views provide information on the attempted rate and the successful connection rate of the initiated tunnels.

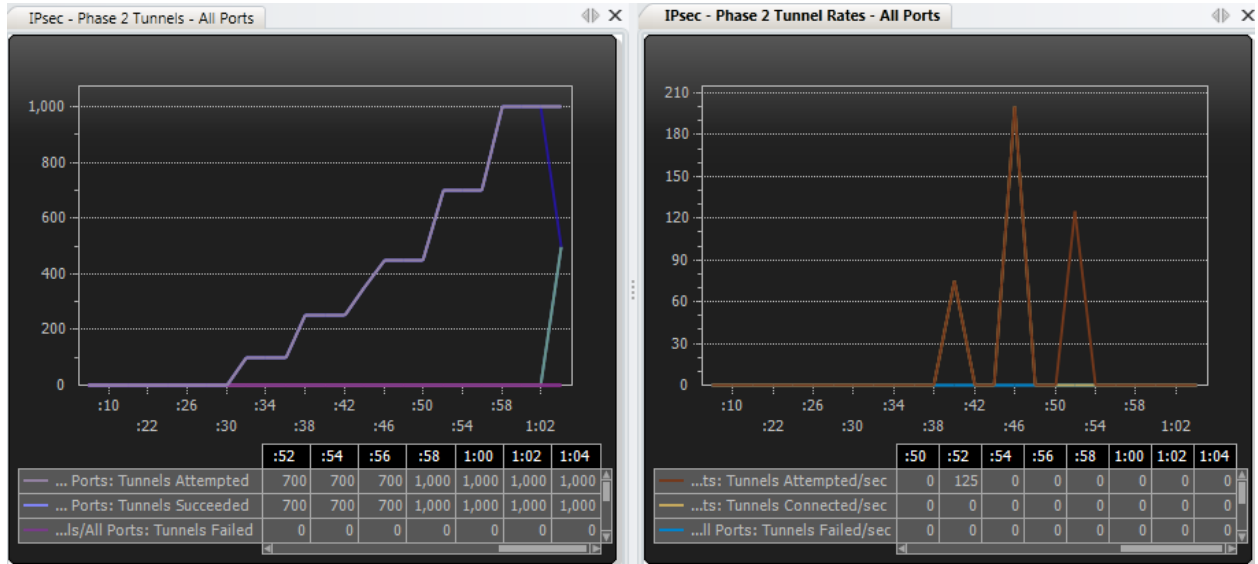


Figure 117. Detailed views for tunnel rate statistics

For detailed results information, generate a Report from the ribbon toolbar by clicking the corresponding report type **PDF Report** or **HTML Report**.

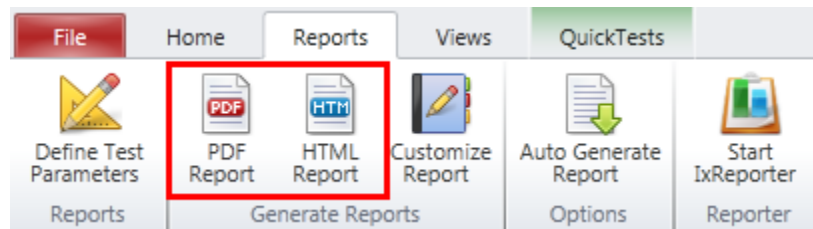
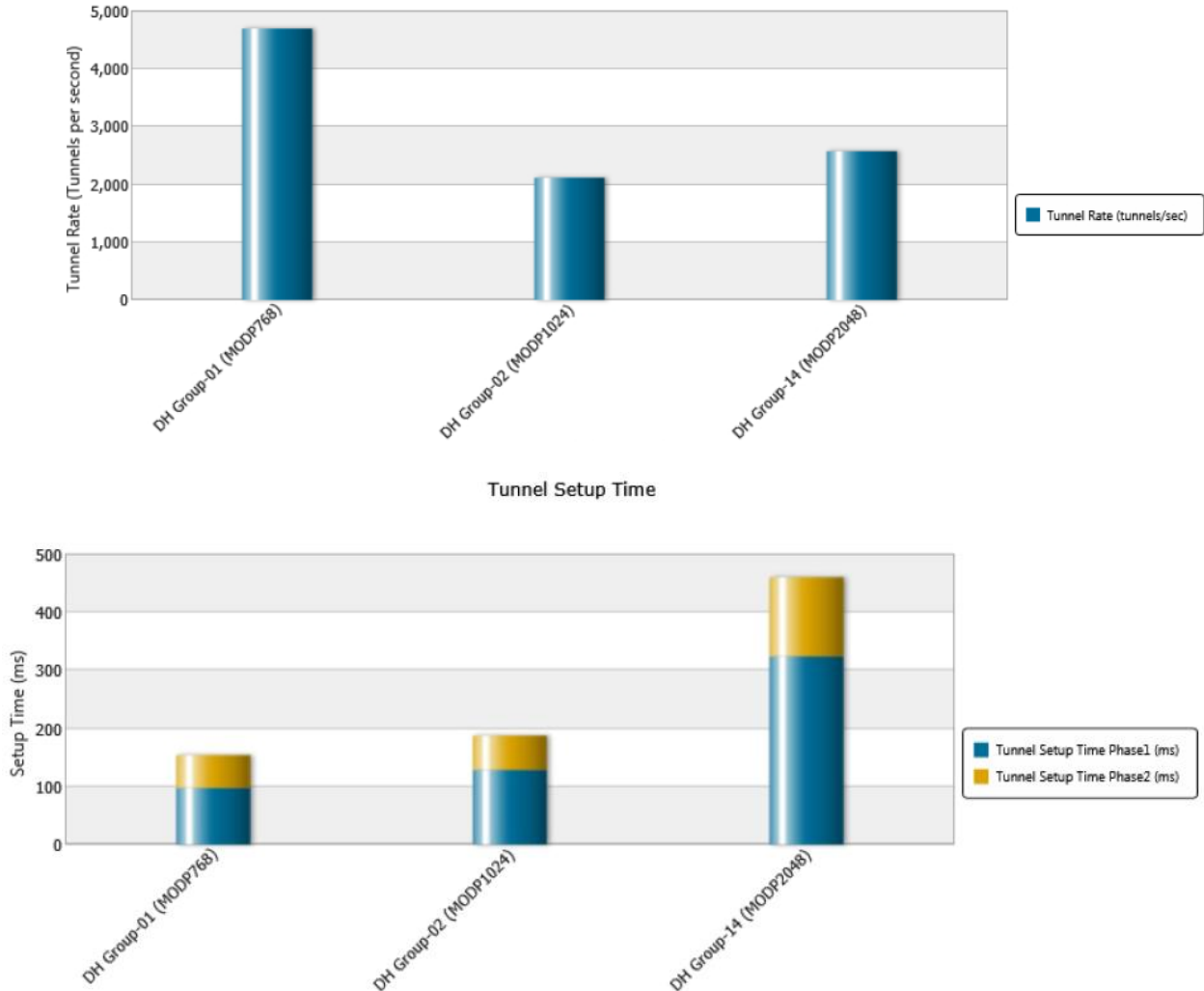


Figure 118. Generating Report for Tunnel Setup Rate results

TEST CASE: IPSEC QUICK TEST – TUNNEL SETUP RATE

You may find examples of the available information in the test report generated below. The tunnel rate performance is aggregated in a single view for the targeted DH groups as along with the minimum and maximum time required to set up the tunnels.

Median Tunnel Rate



Conclusions

This test methodology assists in configuring an IPsec Tunnel Set up Rate Quick Test suite using sweep algorithm to determine the sustained tunnel rate performance of an IPsec VPN Gateway.

Test Case: IPsec Quick Test – Tunnel Capacity

Overview

The IPsec Quick Tests are a set of packaged tests designed to benchmark the performance and capacity of IPsec VPN gateways. You can add, access, run, and customize these tests according to your requirements. These tests are created and stored in the IxLoad configuration file.

The IPsec Tunnel Capacity Quick Test measures the number of concurrent IPsec tunnels that the VPN Gateway (DUT) can sustain with or without data traffic.

The test behavior implies a gradual evolution in a step manner until the configured maximum number of tunnels is established or a defined stop criterion is met. First, the test attempts to establish an initial number of tunnels at a configurable rate. Upon completion of each tunnel, data plane traffic can be optionally started through the respective tunnel. After a configurable time duration, another batch (*Step concurrent tunnels*) of tunnels are initiated at the same rate. Upon completion of each of the new initiated tunnels data plane traffic, if selected, can be started through the new tunnels as well. This process of establishing a new batch of tunnels and optionally starting data plane traffic continues until the acceptable tunnel failure threshold is crossed or the configured maximum number of tunnels is established, whichever comes first. The tunnels are not torn down between iterations and the data traffic once started on a tunnel, if the data plane is enabled, continues until the end of the test.

The maximum number of concurrent IPsec tunnels is dependent on the available memory resources, hence it can be directly impacted by:

- **Authentication Method:** PreShared key, RSA Certificates, EAP (SIM, AKA, TLS, and MD5).
 - PreShared key consumes less memory than Certificates and EAP based authentication.
- **IP version:** IPv4, IPv6.
 - More memory is required for IPv6 based endpoints.
- **Number of host per tunnel** (Site-to-Site scenarios):
 - The memory per tunnel increases with the number of hosts configured per tunnel.

Objective

The goal of this test case is to provide step by step guidance on how to use Ixia's IPsec Tunnel Capacity Quick Test to determine the DUT's maximum concurrent IPsec tunnels it can sustain.

TEST CASE: IPSEC QUICK TEST – TUNNEL CAPACITY

Setup

The test topology consists of a remote access deployment where one Ixia test port emulates 100K IPsec clients connected to the public interface of the DUT (IPsec VPN Gateway) and the second test port emulates the protected IP endpoints located behind the private interface of the DUT.

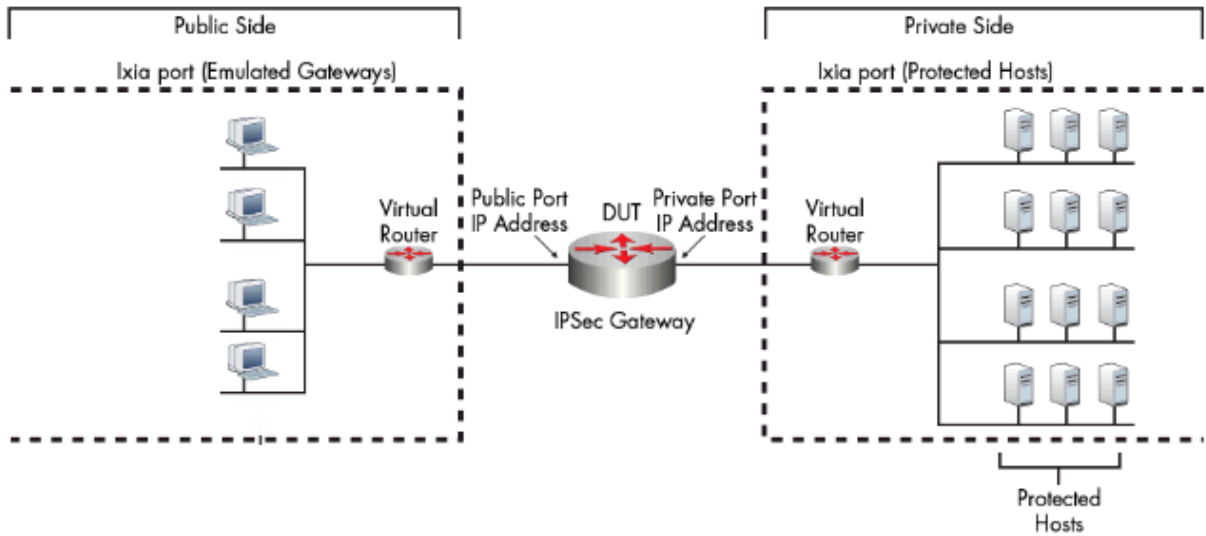


Figure 119. Test Topology

Step-by-step Instructions

1. Start the IxLoad application.
2. Start the quick test framework by selecting **QuickTests** entry from the left-pane.

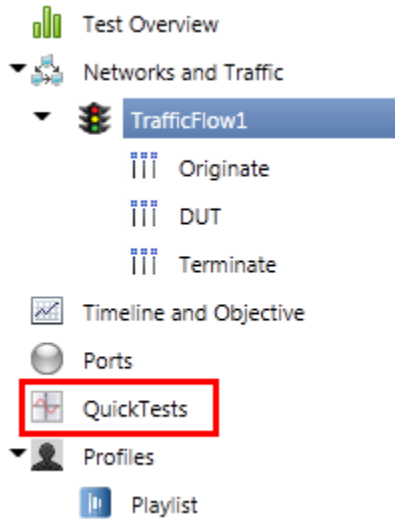


Figure 120. Quick Test Suite

3. Click **Add Quick Test** to add a new quick test. The **Quick Test** wizard starts.

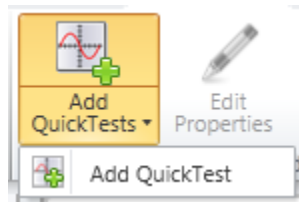


Figure 121. Adding a new test case

TEST CASE: IPSEC QUICK TEST – TUNNEL CAPACITY

- Click **IPsec Tunnel Capacity** and then click **New Configuration** mode:

The **New Configuration** mode builds a configuration from scratch.

The **Existing Configuration** mode is designed to build the test methodology over an existing IxLoad configuration.

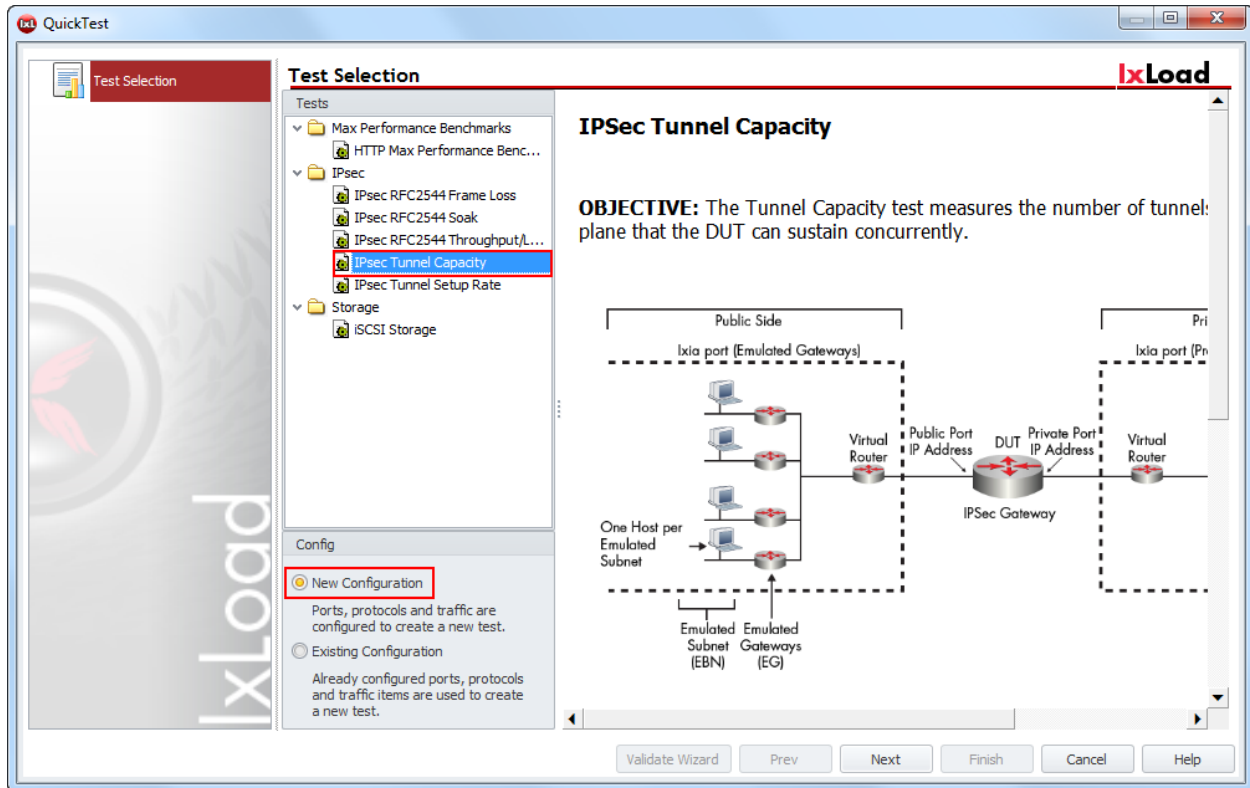


Figure 122. Quick Test suite options

- Select **Port to DUT** test type mode, **Remote Access** scenario and **IKEv2** as the protocol version. This test case methodology uses IKEv2 with single IP range per Network group for the emulated endpoints.

IPsec Configuration

IPsec Test Type Selection <input checked="" type="radio"/> Port to DUT <input type="radio"/> Port to Port	IPsec Test Scenario Selection <input type="radio"/> Site to Site <input checked="" type="radio"/> Remote Access	IKE Version Selection <input type="radio"/> IKEv1 <input checked="" type="radio"/> IKEv2
--	--	---

Unique MACs per EG

Figure 123. Quick Test IPsec test scenarios details

TEST CASE: IPSEC QUICK TEST – TUNNEL CAPACITY

6. Select the appropriate Phase 1 and Phase 2 parameters such as Hash Algorithm, Encryption Algorithm, DH Group, Pre-Shared Key and key Lifetime. Ensure that you configure the same parameters on the DUT to allow a successful tunnel negotiation. Leave the other options to their default values unless changes are necessary.

The image shows two configuration panels for IKEv2. The top panel, 'Phase 1 Options', includes fields for IKE Mode (Main Mode), Hash Algorithm (HMAC-SHA1), Encryption Algorithm (AES-128-CBC), Enable xAuth (unchecked), Seed User Name (ipsec), Seed Password (ipsec), Enable ModeCfg (unchecked), DH Group (MODP-1024 (2)), PRF Algorithm (HMAC-SHA1), Lifetime (sec) (28800), and Pre Shared Key (ipsec). The bottom panel, 'Phase 2 Options', includes fields for AH & ESP Option (ESP Only), Hash Algorithm (HMAC-SHA1), Lifetime (sec) (3600), and Encryption Algorithm (AES-128-CBC).

Phase 1 Options	
IKE Mode	Main Mode
Hash Algorithm	HMAC-SHA1
Encryption Algorithm	AES-128-CBC
<input type="checkbox"/> Enable xAuth	<input type="checkbox"/> Enable ModeCfg
Seed User Name	ipsec
Seed Password	ipsec
<input type="checkbox"/> Enable User Groups	
User Group List	vpngroup
DH Group	MODP-1024 (2)
PRF Algorithm	HMAC-SHA1
Lifetime (sec)	28800
Pre Shared Key	ipsec

Phase 2 Options	
AH & ESP Option	ESP Only
Hash Algorithm	HMAC-SHA1
Lifetime (sec)	3600
Encryption Algorithm	AES-128-CBC

Figure 124. IKEv2 default configuration options

TEST CASE: IPSEC QUICK TEST – TUNNEL CAPACITY

- After setting the security details, configure the network connectivity details according to test environment. The required sets of parameters are highlighted in below figure:

Network Settings

Subnets IP Address Type: IPv4

Gateways IP Address Type: IPv4

Public Area Configuration

Emulated Gateways

First IP Address: 30.0.0.1

Increment By: 0.0.0.1

Range Increment Step: 1.0.0.0

Prefix: 8

Emulated Subnets

First IP Address: 50.0.0.0

Increment By: 0.0.1.0

Range Increment Step: 10.0.0.0

Num Hosts: 1

Prefix: 24

Private Area Configuration

Emulated Gateways

First IP Address: 40.0.0.1

Increment By: 0.0.0.1

Range Increment Step: 1.0.0.0

Prefix: 8

Protected Subnets

First IP Address: 140.0.0.0

Increment By: 0.0.1.0

Range Increment Step: 10.0.0.0

Num Hosts: 1

Prefix: 24

Single Protected Subnet

DUT Configuration

Public IP Address: 22.22.22.22

Private IP Address: 11.11.11.11

Figure 125. Network configuration settings

- Configure the **Emulated Router** networking details. This represents a virtual router emulated on an Ixia port as a network entity routing all the traffic for all the emulated IP addresses. Use this as a next hop Gateway in the DUT routing rules for the emulated IP addresses on the public and private side.

Emulated Router

Emulated Router Configuration

Public Emulated Router

IP Type: IPv4

IP Address: 22.22.22.23

Increment By: 0.0.0.1

Range Increment Step: 0.0.0.2

Prefix: 24

MSS: 1460

Private Emulated Router

IP Type: IPv4

IP Address: 11.11.11.12

Increment By: 0.0.0.1

Range Increment Step: 0.0.0.2

Prefix: 24

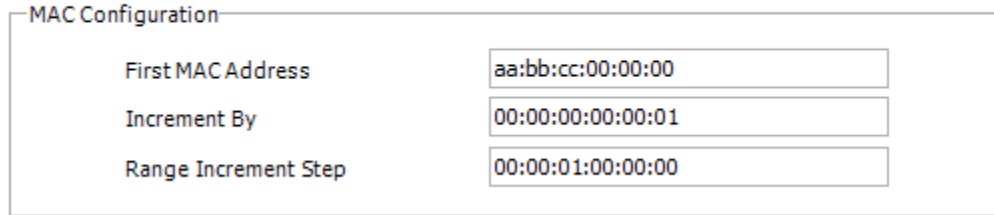
MSS: 1460

Figure 126. Virtual router network details

TEST CASE: IPSEC QUICK TEST – TUNNEL CAPACITY

- Configure the **MAC** addresses if necessary for the Ixia emulated entities.

MAC



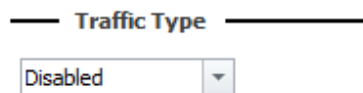
MAC Configuration	
First MAC Address	aa:bb:cc:00:00:00
Increment By	00:00:00:00:00:01
Range Increment Step	00:00:01:00:00:00

Figure 127. MAC Address range options

- Configure the VLAN profiles for the traffic initiated by the Ixia ports. By default, all the traffic is untagged. Leave these options unchanged unless necessary.
- IxLoad's Quick Test offers the option to generate traffic over the established tunnels. By default, the Data Plane option is disabled, which means no traffic is transmitted over the tunnels. There are two options for traffic generation, stateless **UDP** streams with customizable content size and rate, and **HTTP 1.1** traffic for specific page files. However, the data traffic has an impact on the rate at which the tunnels are set up for the device under test.

This test does not use any traffic over the established tunnels, so the traffic type option should remain disabled. However, this option can be enabled, generating a new test case for the DUT performance measurement.

Data Plane Settings



— Traffic Type —

Disabled ▾

Figure 128. Data plane setting options

- Configure the desired Test Parameters for one trial. For this test, we use 100K maximum concurrent IPsec tunnels with a step of 4K new tunnels and a tunnel setup rate of 200 tunnels per second:

Trials: 1. The number of trials that can be run. It may be necessary to run several trials of the tests in order to verify the results for consistency.

Maximum pending tunnels: 100. The maximum number of in progress tunnels at any given time.

TEST CASE: IPSEC QUICK TEST – TUNNEL CAPACITY

Leave tunnels up after test complete: *Cleared*. If selected, the tunnels are not released after the test run is complete

Acceptable Tunnel Failures: *10%*. Percentage/Number of tunnels that are allowed to fail to set up. The test stops if this criterion is met.

Traffic Duration: *30 sec*. The time duration after which next iteration starts once all the tunnels for current iteration are complete.

Tunnel Rate: *200*. The desired rate at which tunnels are established (tunnels/second).

Min concurrent tunnels: *10000*. The initial number of tunnels that a test negotiates in the first iteration.

Max concurrent tunnels: *100000*. The maximum number of concurrent tunnels that a test can negotiate. The test runs (brings up tunnels and starts traffic) as long as the stop criteria is not met or until this number of tunnels is reached.

Step concurrent tunnels: *4000*. The number of tunnels incremented by this amount with each iteration.

Test Parameters **ixLoad**

Test Parameters	Traffic Duration
Trials: <input type="text" value="1"/>	Hours <input type="text" value="0"/> Mins <input type="text" value="0"/> Secs <input type="text" value="30"/>
Maximum pending tunnels <input type="text" value="100"/>	
<input type="checkbox"/> Leave tunnels up after test complete	Iteration Parameters
	Number of Retries <input type="text" value="3"/>
Algorithm Tolerance	Retry interval (sec) <input type="text" value="30"/>
<input checked="" type="checkbox"/> Acceptable Tunnel Failures <input type="text" value="10"/> %	Increment retry interval (sec) <input type="text" value="1"/>
	Tunnel Rate <input type="text" value="200"/>
	Min concurrent tunnels <input type="text" value="10000"/>
	Max concurrent tunnels <input type="text" value="100000"/>
	Step concurrent tunnels <input type="text" value="4,000"/>

Figure 129. Test options for Tunnel Capacity Quick Test

TEST CASE: IPSEC QUICK TEST – TUNNEL CAPACITY

13. Assign the Ixia ports for traffic emulation. The chassis can be the local domain name or the provisioned IP address. After selecting the desired ports, add or remove them to NetTraffics by clicking the corresponding icon.

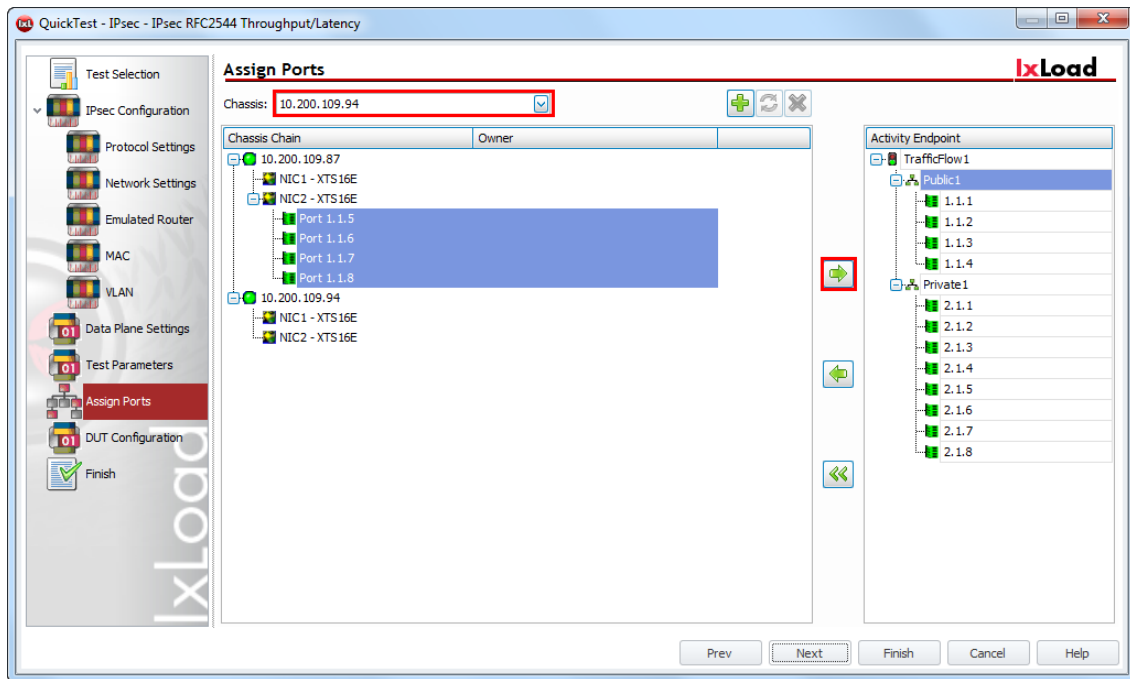


Figure 130. Assigning ports for the emulated networks

14. Click **Next** until the end of the wizard. As the last step, save the configured Quick Test with a suitable name. You can re-use this configuration later for testing.

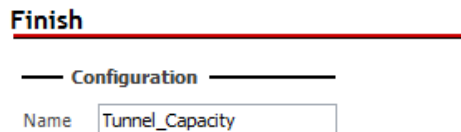


Figure 131. Saving the current Quick Test

TEST CASE: IPSEC QUICK TEST – TUNNEL CAPACITY

15. Select the desired **Quick Test** case. Click **Start** button in the ribbon to start executing the test.

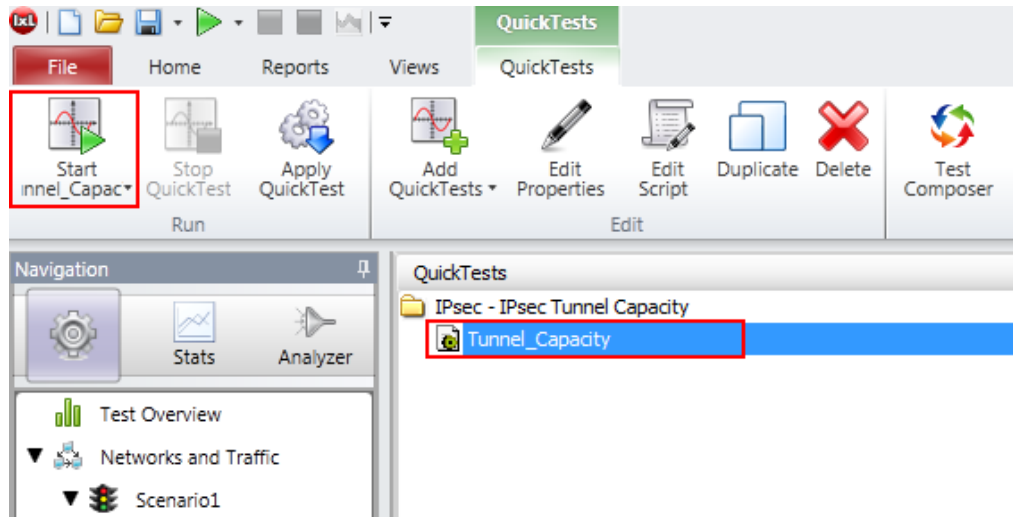


Figure 132. Starting the Quick Test execution

Test Variables

The important test variables impacting the tunnel capacity are as follows:

Parameter Name	Current Value	Comments
Authentication Method	PreSharedKey	<p>Available options:</p> <p>PreSharedKey, RSA Certificates (one per tunnel or one for all tunnels), EAP (SIM, TLS, AKA, MD5)</p>
IP version	IPv4	<p>Available options:</p> <p>IPv4 or IPv6</p>
DH-Group	DH-2	<p>Available options:</p> <p>DH-1(MODP-768), DH-2 (MODP-1024), DH-5 (MODP-1536), DH-14 (MODP-2048), DH-15 (MODP-3072), DH-16 (MODP-4096), DH-17 (MODP-6144), DH-18 (MODP-8192), DH-22 (MODP-1024-5160), DH-23 (MODP-2048-5224), DH-24 (MODP-2048-5256), DH-25 (ECP-192), DH-26 (ECP-224), DH-19 (ECP-256), DH-20 (ECP-384), DH-21 (ECP-521)</p> <p>Recommendation: The smaller DH group, the higher the tunnel rate is. Higher DH group numbers</p>

TEST CASE: IPSEC QUICK TEST – TUNNEL CAPACITY

Parameter Name	Current Value	Comments
		offer better security
Traffic Type	Disabled	Each L4-7 protocol consumes a different amount of memory per user. Experience the results with different application protocols.
Encryption Algorithm	AES-128	Available options: DES, 3DES, AES128, AES192, AES256 (CBC/GSM/GMAC)

TEST CASE: IPSEC QUICK TEST – TUNNEL CAPACITY

Result/Analysis

This section reviews the statistics for the Quick Test Tunnel Capacity:

- Concurrent Tunnels
- Total Tunnels Failed

When performing the test, access the **Quick Test Log** window for runtime status of the test results. The current iteration state and the achieved test results for executed trial are displayed in this log. To view the global test status and eventual warning and errors, access the **Main Log** window, available by default at the bottom of the application window.

The screenshot displays two overlapping log windows. The top window, titled 'Quick Test Log', shows a log of test iterations. The bottom window, titled 'IxLoad Main Log', shows a table of log events.

Quick Test Log Content:

```

8/19/2013 2:26:33 PM:
8/19/2013 2:26:33 PM: =====> iteration 3, trial 1, Tunnels 18000, Started 2:26:33 PM
8/19/2013 2:26:33 PM:
8/19/2013 2:26:33 PM: Setting objective value on the fly: 18000 simulatedUsers
8/19/2013 2:26:33 PM: Waiting for tunnels to establish...
8/19/2013 2:26:55 PM: Done
8/19/2013 2:26:57 PM: Waiting for 30 sec
8/19/2013 2:27:27 PM: Done.
8/19/2013 2:27:27 PM:
8/19/2013 2:27:27 PM:
8/19/2013 2:27:28 PM: Waiting for stats: 2s
8/19/2013 2:27:30 PM: Collecting stats
8/19/2013 2:27:30 PM:
8/19/2013 2:27:30 PM: Results Metrics
8/19/2013 2:27:30 PM: *****
8/19/2013 2:27:30 PM: Trial: 1 Iteration: 3
8/19/2013 2:27:30 PM:
8/19/2013 2:27:30 PM: Concurrent Tunnels  Tunnels Attempted Phase 1  Tunnels Attempted Phase 2  Tunnels Established Phase 1
8/19/2013 2:27:30 PM: *****
8/19/2013 2:27:30 PM: 18000          18000          18000          18000
8/19/2013 2:27:30 PM:
8/19/2013 2:27:30 PM: Total Tunnels Failed (%) : obtained 0 <= 10.
8/19/2013 2:27:30 PM:
8/19/2013 2:27:30 PM: =====> iteration 4, trial 1, Tunnels 22000, Started 2:27:30 PM
8/19/2013 2:27:30 PM:
8/19/2013 2:27:30 PM: Setting objective value on the fly: 22000 simulatedUsers
8/19/2013 2:27:30 PM: Waiting for tunnels to establish...
8/19/2013 2:27:54 PM: Done
8/19/2013 2:27:56 PM: Waiting for 30 sec
    
```

IxLoad Main Log Table:

Time	Severity	Category	Message
Monday, August 19, 2013 2:27:30 PM	Info	Global.Log	Setting objective value on the fly: 22000 simulatedUsers
Monday, August 19, 2013 2:27:30 PM	Info	Community Controller	Changing objective value for agent StatelessPeer1 to 22000 Initiator Peer Count
Monday, August 19, 2013 2:27:30 PM	Info	Global.Log	Waiting for tunnels to establish...
Monday, August 19, 2013 2:27:54 PM	Info	Global.Log	Done
Monday, August 19, 2013 2:27:56 PM	Info	Global.Log	Waiting for 30 sec

Figure 133. Application layout for logging information

TEST CASE: IPSEC QUICK TEST – TUNNEL CAPACITY

Another, more detailed way of monitoring various test statistics in real time is to leverage the available set of IPsec stat views. Click the **Stats** button from the left navigation pane and select from the stat view tree the most relevant statistic views.

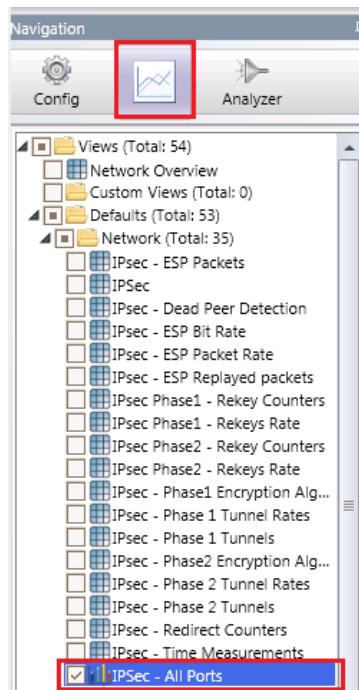


Figure 134. Stat View tree

TEST CASE: IPSEC QUICK TEST – TUNNEL CAPACITY

IPsec – All Ports view provides valuable insights for various KPIs like total number of tunnels initiated, successful and failed, tunnel rate associated statistics (initiated, setup, failed) and even additional DPD and Rekeys related stats. These metrics assist in evaluating the DUT's behavior at different time references corresponding to different loads in terms of established tunnels.

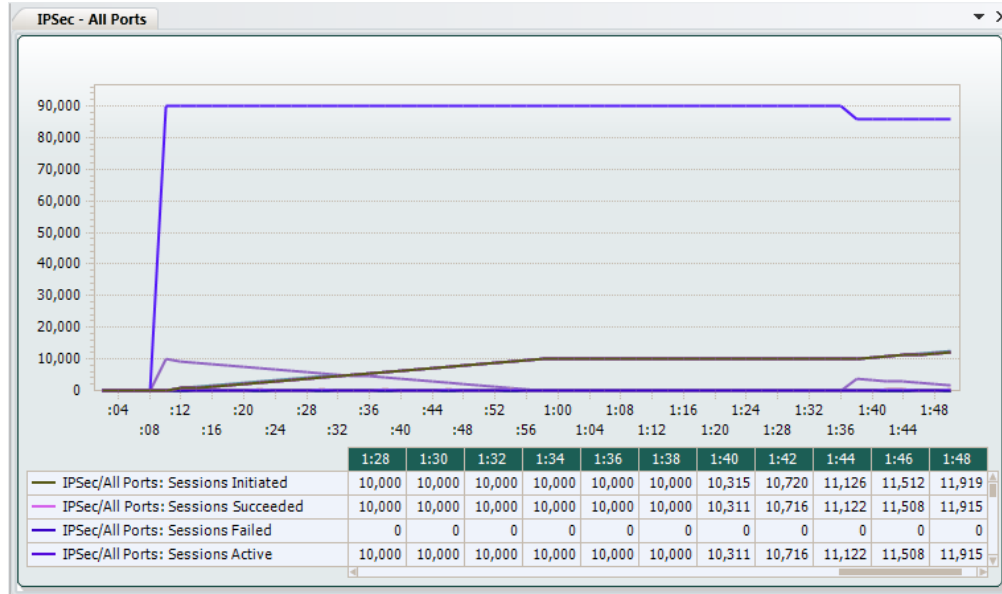


Figure 135. IPsec – All Ports view

IPsec – Time Measurements – All Ports view offers the possibility to monitor the DUT's responsiveness behavior over the entire test duration. It provides statistics like tunnels setup time min, max, and average, but also tunnel teardown time.

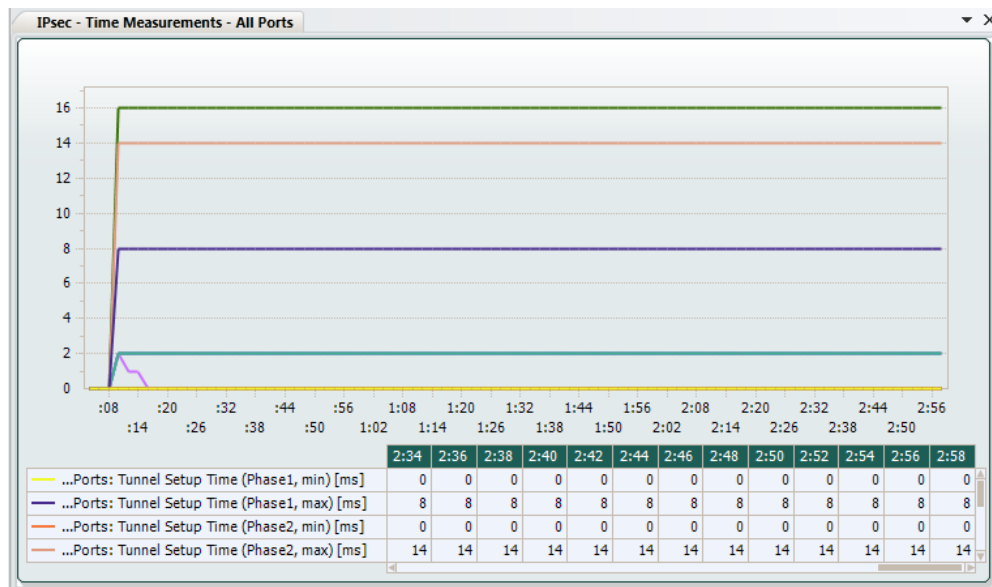


Figure 136. IPsec – Time Measurements – All Ports view

TEST CASE: IPSEC QUICK TEST – TUNNEL CAPACITY

If the number of tunnel failures remains lower than the Acceptable Tunnel Failure threshold, the Tunnel Capacity of the DUT is at least Max concurrent tunnels. A new execution of the test, with a higher Max concurrent tunnels value, must be performed to determine the actual capacity of the DUT. If the tunnel failure exceeds the threshold, the DUT capacity is the number of tunnels established in the last but one step iteration.

At the end of the test, a detailed test report can be generated from the ribbon toolbar by clicking the corresponding report type **PDF Report** or **HTML Report**.

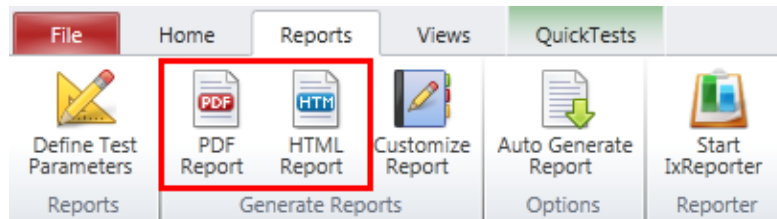


Figure 137. Generating Reports for Tunnel Capacity Quick Test

Conclusions

This test methodology assists in configuring an IPsec Tunnel Capacity Quick Test suite to determine the maximum number of concurrent IPsec tunnels that an IPsec VPN Gateway can sustain within a specified tolerance.

Appendix A: Configuring IP and Network Settings

In the **Scenario Editor**, add a **NetTraffic Object**. This object can contain network configurations and **Activities** (Protocols).



Figure 138. New NetTraffic Object

Click **Network1** to configure the IP, TCP, and other network configuration parameters.

On the IP stack, configure the desired number of static IP addresses. If VLANs are required, configure it by selecting the MAC/VLAN stack and configuring the details.



	Enabled	Name	Status	IP Type	Address	Mask	Increment	Count	Gateway
▶ 1	<input checked="" type="checkbox"/>	IP-R4	Unconfigured	IPv4	192.168.1.2	16	0.0.0.1	1000	192.168.1.1

IP stack

-VLAN

	Enabled	Name	Status	First ID	Increment every # addresses	Increment By	Unique Count
▶ 1	<input checked="" type="checkbox"/>	VLAN-R4	Unconfigured	101	1	1	1000

VLAN settings

Appendix B: Configuring TCP Parameters

The TCP settings shown next should be configured in accordance with the test tool Input Parameters for the specific test case.

Select the **NetTraffic** for the TCP configurations. Select the **Network** object to open the Stack Manager window pane on the bottom.

Click **TCP/IP**. Configure the Receive and Transmit Buffer Size based on what is set in the Input Parameters for TCP/IP settings.

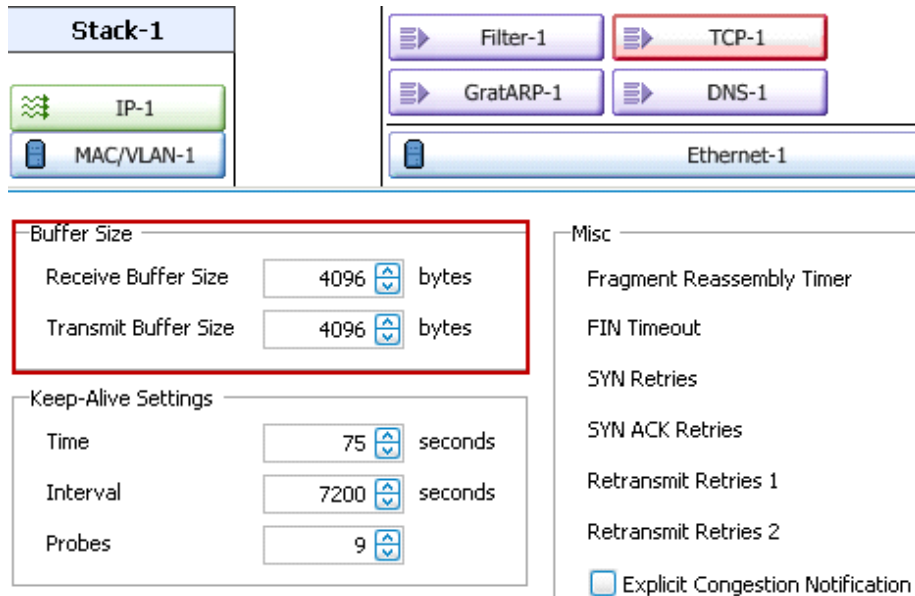


Figure 139. Buffer size settings

Other TCP/IP configurations should not be changed from their defaults.

Appendix C: Configuring HTTP Servers

Add the HTTP server Activity to the server NetTraffic object.

To configure the HTTP server parameters, pages and responses, select the HTTPServer1 object to open the configuration pane at the bottom.

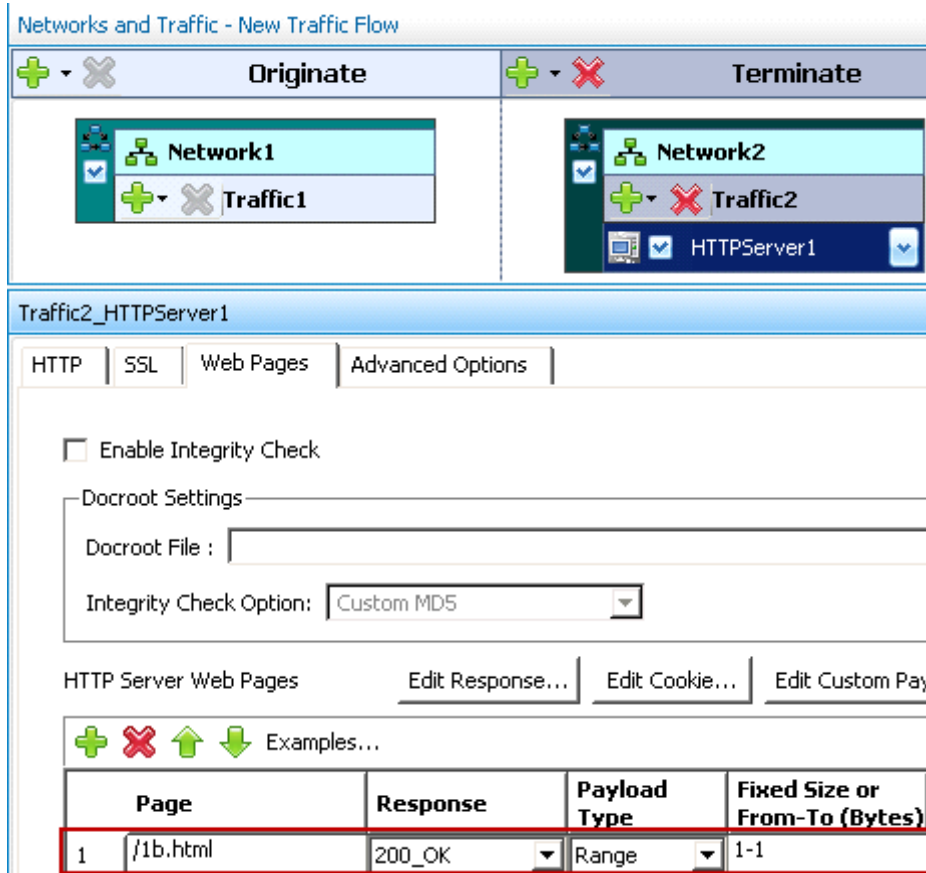


Figure 140. HTTP server configuration

Configure the HTTP server as outlined in the Input Parameters section.

Appendix D: Configuring HTTP Clients

Add the HTTP client Activity to the client NetTraffic object.

To configure the HTTP parameters, and pages to request, select the 'HTTPClient1' object to open the configuration pane on the bottom.

Configure the HTTP behavior on the HTTP tab. Refer to the Input Parameters section.

The version of HTTP to use, TCP connections per user, and transactions per TCP are configured here.

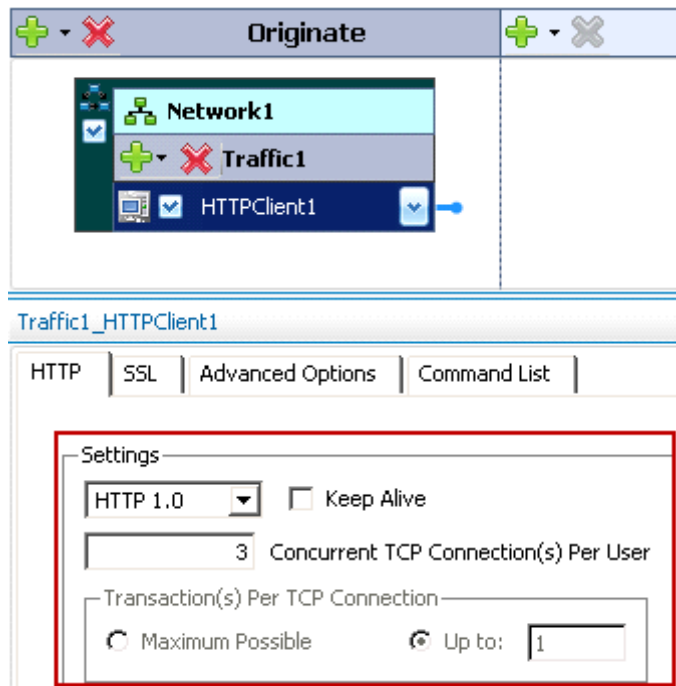


Figure 141. HTTP client configuration

Go to the Command List to configure the list of HTTP commands to use. The specific commands that should be used for the specific test objective type are outlined in the Input Parameters section.

For example, when testing for CPS, the page size is 1b.html.

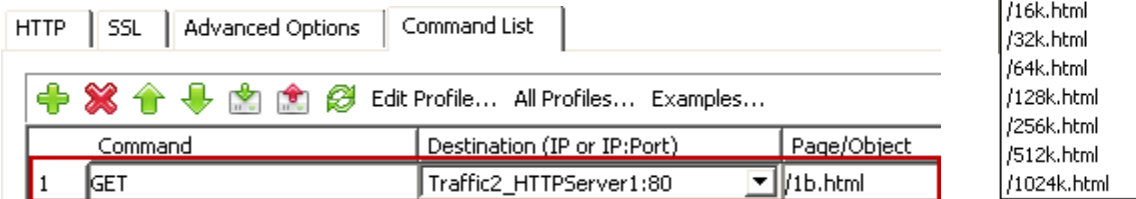


Figure 142. HTTP client command list

APPENDIX D: CONFIGURING HTTP CLIENTS

For throughput testing, the page size is 1024k.html.

Note the Traffic2_HTTPServer1:80 value in the Destination field. IxLoad supports this symbolic destination to associate traffic to the server object. It allows dynamic configuration of traffic distribution from the clients to servers across several ports, without manual configuration.

The use of source IP addresses for a test can depend on the test requirements. To maximize the test tool's performance, this is set to the default configuration of Use Consecutive IPs. This means that every simulated user will use the IPs as needed from the network configuration.

To change it, click the Traffic1 object and change the Use Source IP Rule (per port) setting.

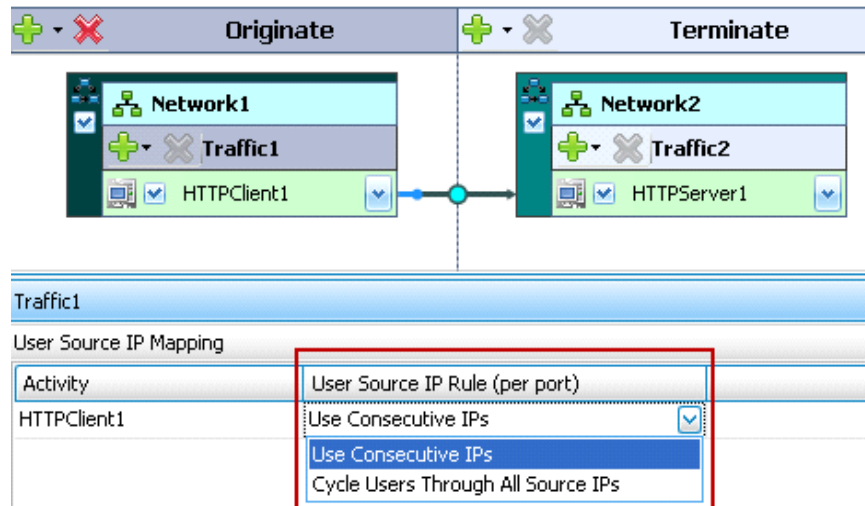


Figure 143. User Source IP Rule (per port)

Using **Cycle Users Through All Source IPs** allows all IP addresses to be used. This may be needed, however, note that performance from the test tool may vary. Consider running a baseline test port-to-port to determine the test tool's limit before performing a test with this feature.

Appendix E: Setting the Test Load Profile and Objective

Open the Timeline and Objective window from the Test Configuration left pane.

Each NetTraffic will be listed, with one or more activities under it. Set the Objective Value to the required value, based on what the target performance is.

Network Traffic Mapping	Objective Type	Objective Value	Timeline
New Traffic Flow			
Traffic1@Network1	Simulated Users	10000	Timeline1
HTTPClient1	Simulated Users	10000	Timeline1
Traffic2@Network2	N/A	N/A	<Match Longest>
HTTPServer1	N/A	N/A	<Match Longest>

Figure 144. Setting the objective value

Set the ramp up and the overall test duration. Use the Input Parameters section as a reference.

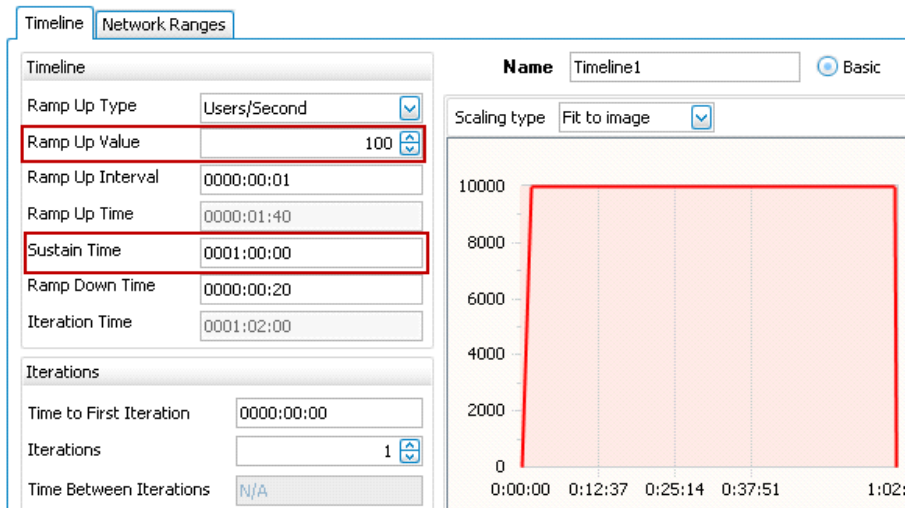


Figure 145. Setting the ramp up and overall test duration

The Number of Ports required highlights the ports required to accomplish the test objective. The computations here are conservative and we recommend you to use Ixia's guidance in addition to the ports required listed here.

Number of Ports Required							
	TXS-128MB	10G-LM	10G-LSM	ELM-1GB	CPM	XMV	ASM XMV12X
Refresh	6	3	2	2	2	2	2

Figure 146. Number of ports required

Appendix F: Adding Test Ports and Running Tests

Go to the Port Assignments window, and add the chassis to be used.

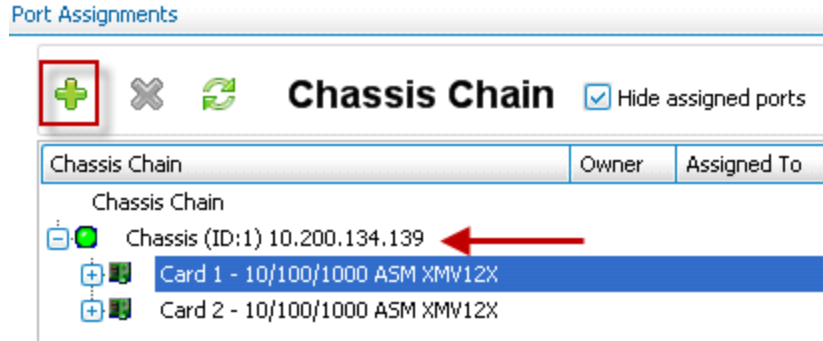


Figure 147. Adding a chassis

The test ports are assigned at the NetTraffic level. In the simplest case in which HTTP client and server traffic is being emulated, there will be two NetTraffics. Add the Required number of ports to each NetTraffic object. Use the arrows to add or remove the available ports to the Assigned Ports.

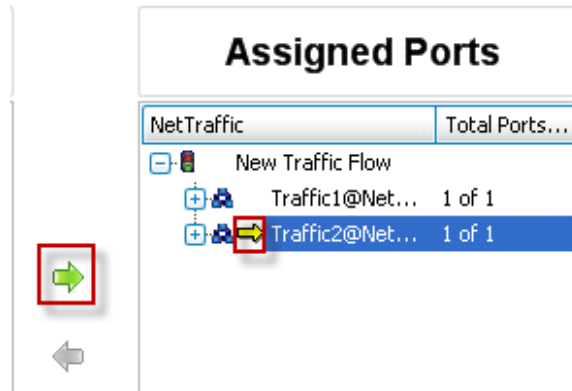





Figure 148. Assigning ports

At this point, the test is ready to be executed.



Figure 149. Executing a test

The PLAY  button starts the test. The RED  button stops the test. The DOWN  arrow downloads the configuration to the test ports, but does not initiate any traffic. The UP arrow de-configures the test ports.

Appendix G: StrongSwan IPsec VPN Gateway Sample Configuration

StrongSwan (www.strongswan.com) is one of the most popular open source IPsec VPN Gateways with a very comprehensive set of IPsec features.

This section includes the StrongSwan configuration file that was used for the IPsec RFC 2544 Throughput and Latency and IPsec Tunnel Setup Rate test methodologies.

```
ipsec.conf config setup
    plutostart=no
    charonstart=yes
    plutodebug=control
    charondebug=dmn4,mgr4,ike4,chk4,job4,cfg4,kl4,net4,enc4,lib4
    strictcrpolicy=no
    crlcheckinterval=0
# Default connection entries

conn %default
    keyingtries=3
    mobike=no
    auto=add
    leftfirewall=yes
# Scripted entries

conn tun1_0_0
    keyexchange=ikev2
    ike=aes128-sha1-modp1024!
    esp=aes128-sha1!
    pfs=no
    right=30.0.0.1
    left=22.20.0.2
    rightsubnet=50.0.0.0/24
    leftsubnet=140.0.0.0/24
    authby=psk
```

Additional modes can be added as comma delimited values for the *ike* variable to allow a more wider range of algorithms to be tested. Example:

```
ike=aes128-sha1-modp1024!, aes128-sha1-modp2048!, aes128-sha1-modp4096!
```

ipsec.secret

```
: PSK "ipsec"
```

By default IxLoad uses the preshared key as "ipsec". This can be changed from Quick Test wizard or from **NetTraffic** -> **IPsec** Stack module -> **Authentication** tab

Appendix H: Application Forwarding Performance under DoS Attacks with Network Impairment Added

Overview

When performing lab network testing, the tester is striving to achieve a realistic reproduction of live networks within the lab. Many times, this test network consists of a good mix of background traffic and protocol test traffic, but the underlying network is pristine and contains no impairments. All production networks contain impairments and the addition of this realism is often overlooked and is the missing link in the creation of a realistic test environment. The addition of Ixia's ImpairNet or Network Emulator impairment devices can bring this realism to the lab.

Objective

Investigate how to implement impairment and add real world randomness that improves the DDoS attack making this attack more realistic. This test is built on the test from the section Application Forwarding Performance Under DDoS Attack, hence this information will not be repeated.

The test implements the following impairments:

- Reorder
- Delay jump
- Accumulate burst

Setup

This test comprises two separate components:

- DDoS Attack as described in section Application Forwarding Performance Under DDoS Attack and not repeated here.
- Network Emulator configured to perform reorder, delay jump, and accumulate burst.

APPENDIX H: APPLICATION FORWARDING PERFORMANCE UNDER DOS ATTACKS WITH NETWORK IMPAIRMENT ADDED

The DDoS Attack test is run two separate times to determine the effects of the impairment configuration. These Tests are:

1. DDoS Attack only to determine baseline. Record this information and use as a base line.
2. Test all three impairments together.

Step by Step Instructions

1. Configure the Network Emulator for reorder, delay jump, and accumulate burst.
2. Log on to the Network Emulator. The following image depicts the Welcome screen.

The screenshot displays the IXIA GEM Network Emulator's web interface. At the top, the IXIA logo is on the left, and a status bar shows 'Blade Tx/Rx' with 3 blades for Transmit and 1 blade for Receive, both at 0.00%/0.00% bandwidth/load. Below the status bar is a 'Welcome' header with a 'Help' link. The main content area is divided into a left navigation menu and a right main panel. The navigation menu includes 'Blade 3 (GEM)', 'Blade 1 (GEM) >>', 'PHY/MAC Control', 'PHY/MAC Stats', 'Manage Profiles', 'Profile Stats', 'Memory Allocation', 'Network Playback', 'Capture/Replay', 'TIA-921/G.1050', 'Chassis Admin', 'System Setup', 'Save/Restore', 'System Info', 'Copy Settings', 'Filter Library', 'Logging', and 'Help'. The main panel displays a 'Welcome to your GEM Network Emulator' message, explaining that GEM is a precision test instrument for Gigabit Ethernet emulation testing, and provides information about its capabilities and control methods. The status bar also shows 'Web Gui: Enabled' and 'Security Control: Disabled'.

APPENDIX H: APPLICATION FORWARDING PERFORMANCE UNDER DOS ATTACKS WITH NETWORK IMPAIRMENT ADDED

3. Click **MANAGE PROFILES** in the left pane. The following window appears. Profile 0 is used for all impairments, because all traffic must be subject to the impairments configured.

The screenshot shows the IXIA Manage Network Profiles web interface. At the top, there is a status bar with the IXIA logo and performance metrics for Blade Tx/Rx (3 and 1), Transmit and Receive bandwidth/load (0.00%/0.00%), and a Help link. Below this is a navigation bar with tabs for Configure Classifier, Bandwidth, Delay/Impairments, Statistics, DSF, and Video. The main content area is titled 'Manage Network Profiles' and shows configuration options for Profile 0: (Default). The profile is set to Blade 1, is enabled, and has the name 'Default'. A 'Filter Rules' section explains that Profile 0 is the default profile and that traffic not matching other filters goes through it. It also notes that each profile can emulate a different network scenario with its own bandwidth, delay, and other impairment settings. A 'Reset All Stats' button is located at the bottom left of the main content area.

Blade Tx/Rx	Transmit	bandwidth/load	Receive	bandwidth/load
3	<input type="checkbox"/>	0.00%/0.00%	<input type="checkbox"/>	0.00%/0.00%
1	<input type="checkbox"/>	0.00%/0.00%	<input type="checkbox"/>	0.00%/0.00%

IXIA

Manage Network Profiles [Help](#)

[Configure Classifier](#) → [Bandwidth](#) → [Delay/Impairments](#) | [Statistics](#) | [DSF](#) | [Video](#)

Blade 3 (GEM)

Blade 1 (GEM) >>

- PHY/MAC Control
- PHY/MAC Stats
- Manage Profiles**
- Profile Stats
- Memory Allocation
- Network Playback
- Capture/Replay
- TIA-921/G.1050
- Chassis Admin
 - System Setup
 - Save/Restore
 - System Info
 - Copy Settings
 - Filter Library
 - Logging
- Help

Profile: 0: (Default) ▼

Blade: 1

Enabled:

Name: Default

Filter Rules

Profile 0 is the default profile.

Traffic that does not match any other profile's filters goes through Profile 0.

Select one of the other available profiles to set up different filters for different types of traffic.

Each profile can emulate a different network scenario, with its own bandwidth, delay and other impairment settings.

APPENDIX H: APPLICATION FORWARDING PERFORMANCE UNDER DOS ATTACKS WITH NETWORK IMPAIRMENT ADDED

4. Reorder

Select PROFILE 0. Click **Delay/Impairments** link as depicted in the image. In the **Packet Impairments** section, select **Reorder** check box. Accept the default values of 1 in 10 **Packets** and click **Apply**.

The screenshot displays the IXIA Network Profile Impairments configuration page. The interface includes a sidebar on the left with navigation options such as 'Blade 3 (GEM)', 'Blade 1 (GEM)', and 'Manage Profiles'. The main content area is titled 'Network Profile Impairments' and features a breadcrumb trail: 'Configure Classifier -> Bandwidth -> Delay/Impairments'. The 'Delay and Jitter' section is active, showing 'Delay Variation' options: 'None (Constant Delay)' (selected), 'Gaussian', 'Internet', 'Uniform (Sawtoothed)', and 'Uniform (Uncorrelated)'. A graph labeled 'No Jitter' shows a vertical blue line. Below the graph, 'Delay Units' is set to 'km' and 'Delay Avg' is '0.198 km'. The 'Packet Impairments' section is expanded, showing 'Reorder (enabled)' with 'Select: 1 in 10 Packets', 'Repeat: Forever', and 'Packet offset range: 1 to 5 packets'. Other impairment options like 'Drop', 'Duplicate', 'AccumulateBurst', 'Modify', 'Corrupt', 'CRCCorrupt', and 'IPv4Fragment' are listed as disabled. Buttons for 'Apply', 'Revert', and 'Reset' are at the bottom.

APPENDIX H: APPLICATION FORWARDING PERFORMANCE UNDER DOS ATTACKS WITH NETWORK IMPAIRMENT ADDED

5. Delay Jump

Click **Delay/Impairments** as depicted in the image. In the **Delay and Jitter** section, click **Gaussian**. Enter 100 in the **Delay Max** text box. Click **Apply**.

The screenshot shows the IXIA Network Profile Impairments configuration interface. At the top, there are status indicators for Blade Tx/Rx (3 Transmit, 1 Receive) and bandwidth/load (0.00%/0.00%). The main navigation bar includes 'Configure Classifier', 'Bandwidth', 'Delay/Impairments', 'Statistics', 'DSF', and 'Video'. The left sidebar lists various system and network management options, with 'Manage Profiles' highlighted. The main content area is titled 'Network Profile Impairments' and shows the configuration for 'Profile #0: (Default)'. Under the 'Delay and Jitter' section, the 'Gaussian' radio button is selected. A graph displays a Gaussian distribution curve. The 'Delay Units' are set to 'ms', and the 'Delay Max' is set to '100.00000 ms'. Below the graph, there is a 'Packet Impairments' section with several options, all of which are currently disabled: Drop, Duplicate, Reorder, AccumulateBurst, Modify, Corrupt, CRCCorrupt, and IPv4Fragment. The 'Reorder' option is highlighted. At the bottom of the configuration area, there are 'Apply', 'Revert', and 'Reset' buttons.

APPENDIX H: APPLICATION FORWARDING PERFORMANCE UNDER DOS ATTACKS WITH NETWORK IMPAIRMENT ADDED

6. Accumulate Burst

Click **Delay/Impairments** as depicted in the image. In the **Packet Impairments** section, select the **AccumulateBurst** checkbox. In the **Accumulation Mode** drop-down list, select **N Packets And Timeout**. Set the **Burst Size** to 100. Click **Apply**.

The screenshot displays the IXIA Network Profile Impairments configuration page. The left sidebar contains navigation options for Blade 3 and Blade 1, including PHY/MAC Control, PHY/MAC Stats, Manage Profiles, Network Playback, Capture/Replay, TIA-921/G.1050, Chassis Admin, System Setup, Save/Restore, System Info, Copy Settings, Filter Library, Logging, and Help. The main content area is titled 'Network Profile Impairments' and includes a breadcrumb trail: 'Configure Classifier → Bandwidth → Delay/Impairments | Statistics | DSF | Video'. The 'Delay and Jitter' section is currently selected, showing 'Delay Variation' options: 'None (Constant Delay)' (selected), 'Gaussian', 'Internet', 'Uniform (Sawtoothed)', and 'Uniform (Uncorrelated)'. A graph labeled 'No Jitter' shows a vertical blue arrow. Below the graph, 'Delay Units' is set to 'km' and 'Delay Avg' is 0.198 km. The 'Packet Impairments' section is expanded, showing 'AccumulateBurst (enabled)' with 'Accumulation Mode' set to 'N Packets And Timeout', 'Burst Size (N)' set to 100, and 'Timeout' set to 1.000000 ms. Other impairment options are disabled. Buttons for 'Apply', 'Revert', and 'Reset' are located at the bottom of the configuration area.

7. Run Combined Test

Test Variables

Test Tool Variables

Reorder Test tool variables

Parameter	Description
-----------	-------------

APPENDIX H: APPLICATION FORWARDING PERFORMANCE UNDER DOS ATTACKS WITH NETWORK IMPAIRMENT ADDED

Reorder	1 in 10 – Selects 1 in 10 packets to be reordered.
Distribution	Periodic – Causes the spacing to be fixed.
Repeat	Forever – Causes test to repeat until the test is stopped.
Packet offset range	1 to 5 – Defines the range of valid reorder.

Delay Test tool variables

Parameter	Description
Delay Variation	Gaussian – Selects the type of variation in the delay.
Delay Avg	100 ms

Accumulate Burst Test tool variables

Parameter	Description
Accumulation Mode	N Packets and Timeout – Selects the burst mode triggered on either N Packets or the Timeout triggers.
Burst Size	100 – Size of the burst.
Timeout	1.0000 - Time out interval.

Conclusions

Network Emulation is normally the missing link to realistic network testing and is often ignored in the overall test planning. The procedures described above show a simple method of adding basic impairments to the Network Emulator. After adding Network Emulation to the network test system, the network reflects a real world environment and reflects better conditions that are found when the product is deployed. The addition of Ixia's ImpairNet or Network Emulator impairment devices can bring this realism to the lab.

Contact Ixia

Corporate Headquarters
Ixia Worldwide Headquarters
26601 W. Agoura Rd.
Calabasas, CA 91302
USA
+1 877 FOR IXIA (877 367 4942)
+1 818 871 1800 (International)
(FAX) +1 818 871 1805
sales@ixiacom.com

Web site: www.ixiacom.com
General: info@ixiacom.com
Investor Relations: ir@ixiacom.com
Training: training@ixiacom.com
Support: support@ixiacom.com
+1 877 367 4942
+1 818 871 1800 Option 1 (outside USA)
online support form:
<http://www.ixiacom.com/support/inquiry/>

EMEA
Ixia Technologies Europe Limited
Clarion House, Norreys Drive
Maiden Head SL6 4FL
United Kingdom
+44 1628 408750
FAX +44 1628 639916
VAT No. GB502006125
salesemea@ixiacom.com

Renewals: renewals-emea@ixiacom.com
Support: support-emea@ixiacom.com
+44 1628 408750
online support form:
<http://www.ixiacom.com/support/inquiry/?location=emea>

Ixia Asia Pacific Headquarters
21 Serangoon North Avenue 5
#04-01
Singapore 5584864
+65.6332.0125
FAX +65.6332.0127
Support-Field-Asia-Pacific@ixiacom.com

Support: Support-Field-Asia-Pacific@ixiacom.com
+1 818 871 1800 (Option 1)
online support form:
<http://www.ixiacom.com/support/inquiry/>