Auditing Fragrouter-1.6 (Vulnerability Test Tool):
An Auditor's Perspective

Auditing Networks, Perimeters, and Systems
GSNA Practical Assignment
Version 2.1 – Option 1

NAM KYUN BAIK

May 11, 2004

Assignment 1 – Research in Audit, Measurement Practice, and Control

1.1 – Abstract

As each network builds up network that can share a variety of information, the whole world can communicate with each other, because of the rapid propagation of Internet environment using TCP/IP. This acceleration of information bestows favor that can share necessary information on ISPs (information service provisions). On the other hand, each network admits private and social important information to be illegally intruded and attacked because the technical aspects of information protection have not kept up.

As in correspondent network security methods, a use for NIDS (Network-based Intrusion Detection System) is gradually increased, hereupon accredited laboratories inform a basis of evaluation of NIDS and arrange a institution of evaluation so that user may use safely reliable NIDS.

Evaluation team about NIDS operates assess course such as development course, testing course, configuration management, operation environment, an explanatory note, vulnerability analysis, then done evaluation. Among these courses, vulnerability analysis is the most important step for delegating weakness NIDS itself. In addition, automatic testing tool can be used or developed for efficient vulnerability analysis on a NIDS. In real environment for evaluating any NIDS, it is important that testing tool is operating correctly and appropriately as specified or desired demands. If vulnerability analysis testing tool is configured improperly and its function does not carry out correctly, the result of test can not be reliable as well as an auditor can not detect flaws on the NIDS. Consequently, asset to be protected by NIDS can expose to the attacker.

An objective of functional test using the vulnerability testing tool is to counter the risk of an incorrect assessment of the test outcomes about NIDS. Therefore, the purpose of this paper is to discuss the auditing steps and procedures on the vulnerability analysis testing tool(Fragrouter-1.6) itself.

1.2 – Identify the System to be Audited

If IP has a datagram to send, and the datagram is larger than the link layer's MTU, IP performs fragmentation, breaking the datagram up into smaller pieces, so that each fragment is smaller than the MTU. When an IP datagram is fragmented, it is not reassembled until it reaches its final destination. The IP layer at the destination performs the reassembly. While the goal is to make fragmentation and reassembly transparent to the transport layer (TCP and UDP) with performance efficient, addressing fragmentation has proven to be rather problematic from a security perspective. Because many NIDS do not adequately deal with IP fragmentation and reassembly. Fragmentation technique to avoid detection by NIDS has been gaining in popularity.

Fragrouter-1.6 (network intrusion detection evasion toolkit) is a program for vulnerability analysis test about NIDS, according to the specific TCP/IP evasion attack. It can fragment and route TCP/IP packet through Internet in order to elude most NIDS. If a NIDS have not function which fragmented packets can be reassembled, attack is success and assets to be protected are exposed. In conclusion, Fragrouter is a uni-direction fragmentation router. When IP packet is transmitted to Fragrouter from attacker, Fragrouter covert a fragmented data stream (various evasion attack methods) and forward to the victim system. Therefore, Fragrouter can be used vulnerability analysis test to the NIDS.

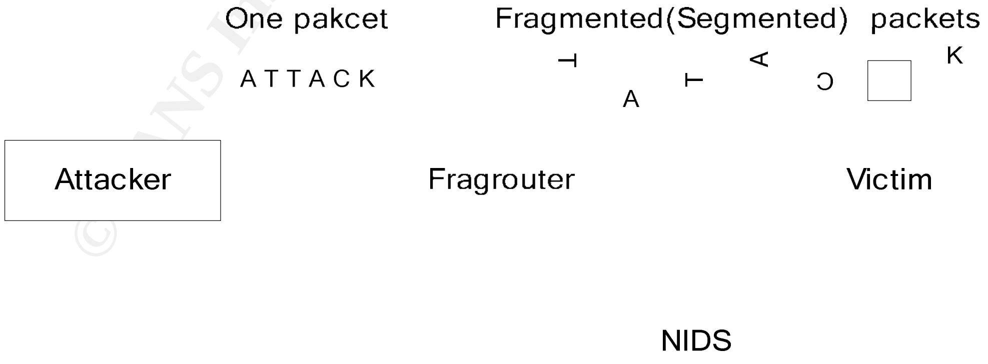


Figure 1. A concept of Fragrouter

The Fragrouter that will be audited is a IBM desktop PC with an Intel Pentium III CPU running at 2.8 Ghz and 512MB of physical RAM. It also has two 10/100 Mbps Network Interface Cards (NIC) and 80GB hard disk space.

Table 1. Overview about Fragrouter system

Tool Name	Fragrouter
Tool Version	1.6
Role	network intrusion detection evasion toolkit
O/S Flatform	Hancom Linux 2.2.1
CPU	P-III 2.8G
RAM	512M
HDD	80G
NIC	10/100 Ethernet Card □ 2

1.3 – Evaluate the Risk to the System

Understanding the relationship between risk and control is important for information security system auditor. They must be able to identify and differentiate risk types and the controls used to mitigate these risks. They must also be able to make assessments of risk to help focus and plan audit work.

One of the most succinct definitions of risk used within the information security business world is provided by the Guidelines for the Management of IT Security published by the International Organization for Standardization (ISO):

> "The potential that a given threat will exploit vulnerabilities of an asset or group of assets to cause loss or damage to the assets. The impact or relative severity of the risk is proportional to the business value of the loss/damage and to the estimated frequency of the threat."

Risk analysis is a process to determine the exposures and their potential harm. Above all, all threat of a target system are listed and explained. Then, for each threat, effects and damages should be analyzed. The last step of the analysis is the establishment of possible control to reduce affect of a threat. Consequently, risk analysis leads to a security plan, which identified responsibility for certain actions to improve security.

This chapter will focus on identified risks that are directly related result from improper operation and environment of the target system - Fragrouter.

· Identification of Assets

While asset can be defined as information or resources – data, hardware, or software - to be protected by target system, in this paper, it means that information or resources is protected NIDS tested by Fragrouter. Therefore, to protect assets, Fragrouter should be correctly working as a given functionality in intended environment.

· Identification of a Risk to the Assets

There are two categories of risk to the Fragrouter which will be examined throughout this paper. One is related to the environment which NIDS's vulnerability is analyzed by Fragrouter, including known physical, personal, procedural security and network configuration. The other one is the failure of function which the functionality can not exhibit the properties necessary to satisfy the functional requirements.

· Threat, Likelihood, Effect and Control

Category : **Environment**
What Can Go Wrong (Threat) : Physical security

How Likely is it to Happen (Likelihood) : An unauthorized user to the physical access can enter to areas containing the Fragrouter (during normal working hours and at other times).

What are the Consequences (Effect) : An unauthorized user can do theft or deliberate damage physically to the system which Fragrouter was installed and physical environment.

How to control (Countermeasure) : Physical access to the areas which Fragrouter is located should be restricted through physical security measures.

Category : **Environment**
What Can Go Wrong (Threat) : Procedural security

How Likely is it to Happen (Likelihood) : Fragrouter might be installed improperly and set-up in unsecured manner.

What are the Consequences (Effect) : Fragrouter can not be performed as specified functionality. Therefore, vulnerability analysis result to the NIDS using by Fragrouter can not be reliable.

How to control (Countermeasure) : During the Fragrouter is installed and set-up, tester shall comply with predefined procedures and regulations.

Category : **Environment**
What Can Go Wrong (Threat) : Personnel security

How Likely is it to Happen (Likelihood) : If a tester was not educated appropriately or did not have sufficient knowledge and skill, he can not operate the Fragrouter as specified functionality and may take a mistake or misuse.

What are the Consequences (Effect) : Fragrouter can not be performed as specified functionality. Therefore, vulnerability analysis result to the NIDS using by Fragrouter can not be reliable.

How to control (Countermeasure) : To minimize the probability of mistake and misuse about the Fragrouter, the tester should be educated properly to learn sufficient knowledge and skill necessary for the operation of the Fragrouter and system.

Category : **Environment**

What Can Go Wrong (Threat) : Network configuration security

How Likely is it to Happen (Likelihood) : If network configuration is improper, evasion packets of Fragrouter can not be transferred through network for the target system.

What are the Consequences (Effect) : The tester can not derive a completed and correct result from the test. In addition to, evasion packets can cause attack or trouble to the other system that is not relevant to the test.

How to control (Countermeasure) : The tester shall prepare network configuration diagram that can verify independent test environment and confirm that network is configured appropriately.

Category : **Functionality**

What Can Go Wrong (Threat) : Fragmentation(Segmentation) attacks

How Likely is it to Happen (Likelihood) : By program error or other reasons, function which is performed by Fragrouter may not be satisfied to intended demands.

What are the Consequences (Effect) : All results which are tested are incorrect and unreliable. So, tested NIDS can not protect assets and exposure may be occurred.

How to control (Countermeasure) : Tester shall provide documents that can verify the consistency between expected result and actual.

1.4 – Current State of Practice

Though searching firsthand resources about auditing Fragrouter on the internet, I could not find any valuable information. But, there are many open materials that are related indirectly. So, I create new audit methodology based on personal experiences and reference materials. The following resources will be used to audit on the Fragrouter.

1.4.1 Research and Documentation:

The following sources have been consulted.

- "Guidelines for the Management of IT Security" published by the International Organization for Standardization (ISO)
- "Insertion, Evasion, and Denial of Service: Eluding Network Intrusion Detection," available via at http://www.securityfocus.com/data/library/ids.ps
- Manpage of Fragrouter (*root#man fragrouter*)
- Manpage of ps (*root#man ps*)
- Manpage of top (*root#man top*)
- Manpage of sar (*root#man sar*)
- Manpage of ifconfig (*root#man ifconfig*)
- "50 Ways to Defeat Your Intrusion Detection System, " available via at http://all.net/journal/netsec/1997-12.html
- "Intrusion Detection FAQ - The Internet's most trusted site for vendor neutral intrusion detection information," available via at http://www.sans.org/resources/idfaq/index.php

- "Multiple Levels of De-synchronization and other concerns with testing an IDS system," available via at http://www.securityfocus.com/infocus/1204

- "Resynchronizing NIDS Systems," available via at http://www.securityfocus.com/infocus/1226

- "IDS Evasion with Unicode, " available via at http://www.securityfocus.com/infocus/1232

- "IDS Evasion Techniques and Tactics," available via at http://www.securityfocus.com/infocus/1577

- "Social Engineering," available via at http://www.securityfocus.com/infocus/1229

- "IDS Infosec Archive," available via at http://www.securityfocus.com/infocus/ids

- "NIST Special Publication on Intrusion Detection Systems," available via at www.21cfrpart11.com/files/library/government/intrusion_detection_systems_0201_draft.pdf

1.4.2 Tools

The following tools have been used.

- "Fragrouter source package," available via at http://packages.qa.debian.org/f/fragrouter.html

- "Analyzer (Packet Sniffering Tool)," available via at http://analyzer.polito.it/

- "Hailstorm V1.2," available via at http://www.securityfocus.com/products/1367

- "Whisker," available via at http://www.securityfocus.com/guest/670

- "IDSwakeup," available via at http://www.securityfocus.com/tools/1803

Assignment 2 - Create an Audit Checklist

2.1 Checklist Coverage & Depth Analysis

The auditing Fragrouter is composed of two parts: the environment and the functionality. Each is mapping to one or more audit items and item has a peculiar purpose. In this chapter, the coverage and depth of checklist are explained as shown below.

· Coverage and Depth for auditing about Fragrouter

Category	Item ID	Goal
Environment	Item ID - 1	This item is concerned with physical security measures that used to protect the testing environment.
	Item ID - 2	This item is concerned with procedural security measures that are useful for ensuring that the Fragrouter has been installed and set-up in a secure manner as intended by the tester.
	Item ID - 3	This item investigates whether the Fragrouter can be used in a manner that is improper but that a tester of the Fragrouter would reasonably believe to be correct. And it is to minimize the probability of misusage on the Fragrouter.
	Item ID - 4	The goal of this item is to determine whether network was configured properly in its intended environment.
Functionality	Item ID - 5 ~ 24	The goal of these items are to determine whether the Fragrouter can exhibit the properties necessary to analyze the vulnerability of NIDS.

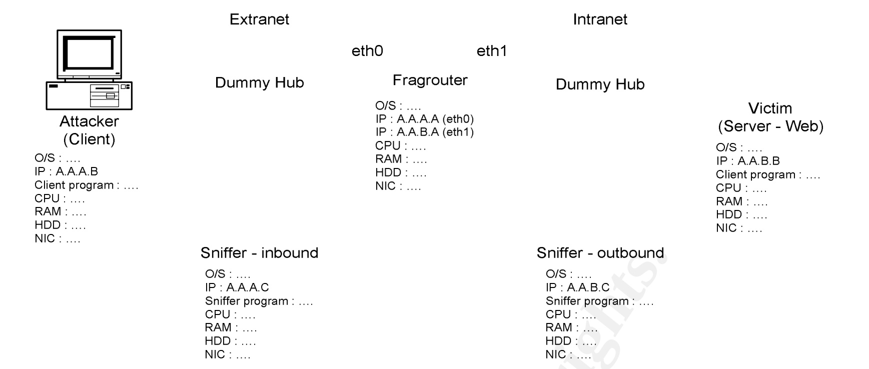
2.2 Checklist for Auditing about Fragrouter

Item ID - 1	**Physical security**
Reference	Personal experience
Control objective	This item is concerned with physical security measures that used to protect the testing environment.
Risk	Poorly controlled access of the testing location can result in vulnerabilities in the physical security. For example, an unauthorized tester who is not responsibility for testing can do theft or deliberate damage to the testing environment.
Compliance	· The tester shall produce physical security documentation. · The physical security documentation shall describe all the physical security measures that are necessary to protect confidentiality and integrity in its testing environment. · The physical security documentation shall provide evidence that these security measures are followed during the testing.
Testing	· step 1. The auditor shall confirm that the information provided meets all requirements for content and presentation of evidence. · Step 2. The auditor shall confirm that the physical security measures are being applied.
Objective / Subjective	Subjective – This is based on the security policy of evaluator or organization.
Expected result	Not applicable
· ***To be completed after the test***	
Actual result	
Audit result	

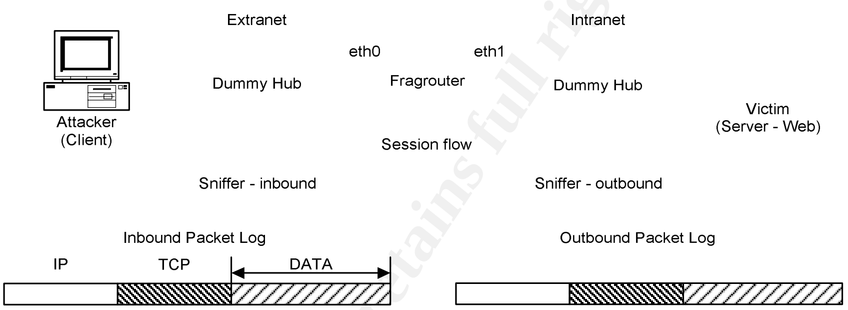
Item ID - 2	Procedural security
Reference	Personal experience
Control objective	This item is concerned with procedural security measures that are useful for ensuring that the Fragrouter has been installed and set-up in a secure manner as intended by the tester.
Risk	If Fragrouter is not installed and set-up in a secure manner, it can not be performed as specified functionality. Therefore, vulnerability test result to the NIDS can not be reliable.
Compliance	· The tester shall document procedures necessary for the secure installation and set-up of the Fragrouter on the system. · The procedural security documentation shall describe the steps necessary for the secure installation and set-up of the Fragrouter on the system.
Testing	· step 1. The auditor shall confirm that the information provided meets all requirements for content and presentation of evidence. · Step 2. The auditor shall determine that the installation and set-up procedures result in a secure configuration of the system.
Objective / Subjective	Subjective – This is based on the security policy of evaluator or organization.
Expected result	Not applicable
· ***To be completed after the test***	
Actual result	
Audit result	

Item ID - 3	Personnel security
Reference	Personal experience
Control objective	This item investigates whether the Fragrouter can be used in a manner that is improper but that a tester of the Fragrouter would reasonably believe to be correct. And it is to minimize the probability of misusage on the Fragrouter.
Risk	If a tester was not educated appropriately, he can not operate the Fragrouter as specified functionality. Therefore, vulnerability test result to the NIDS can not be reliable.
Compliance	· The tester shall provide evidence that he was educated properly or has sufficient knowledge and skill necessary for the operation of the Fragrouter, network, and system.
Testing	· step 1. The auditor shall investigate whether a tester has been educated or has sufficient knowledge and skill necessary for the operation of the Fragrouter, network, and system. · step 2. The auditor shall interview a tester to confirm that he is a well-qualified man for the position.
Objective / Subjective	Subjective – This is based on the security policy of evaluator or organization.
Expected result	Not applicable
· ***To be completed after the test***	
Actual result	
Audit result	

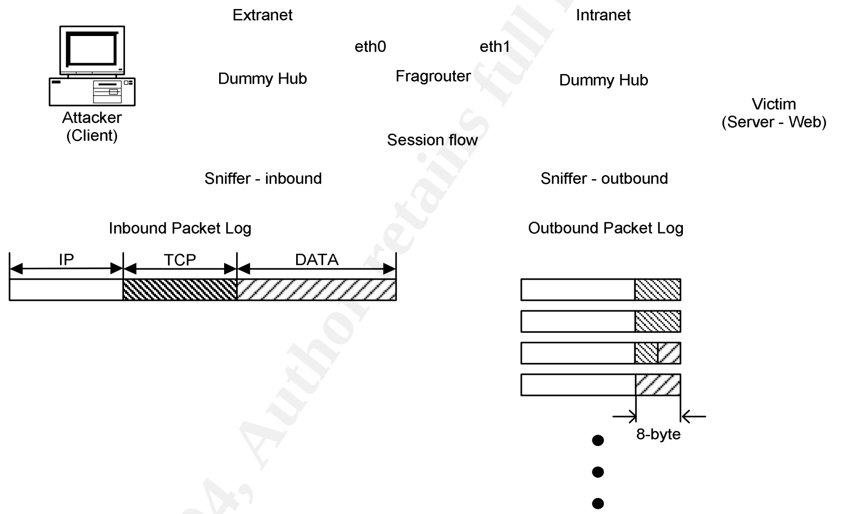
Item ID - 4	Network configuration security
Reference	Personal experience
Control objective	The objective of this item is to determine whether network was configured properly in its intended environment.
Risk	If network configuration is improper, evasion packets can not be transferred through network for the target system. So, tester can not derive a completed and correct result from the test. In addition to, evasion packets can cause attack or trouble to the other system that is not relevant to the test.
Compliance	· The tester shall provide network configuration diagram that can verify independent test environment and confirm that network is configured appropriately.
Testing	· step 1. The auditor shall determine whether test environment is independent or not. - Does Fragrouter take a role of gateway between two different networks ? - Is there no any system that is unrelated to test environment ? · step 2. The auditor shall determine whether IP address is properly assigned to the system or not. - Is assigned IP address valid ? - Is there any duplication of IP address ?
Objective / Subjective	Objective – Results are generated repeatedly.
Expected result	Following is the preferred network environment configuration that is used for vulnerability analysis test on the NIDS.

	 Extranet Intranet eth0 eth1 Dummy Hub Fragrouter Dummy Hub Attacker (Client) O/S : …. IP : A.A.A.B Client program : …. CPU : …. RAM : …. HDD : …. NIC : …. O/S : …. IP : A.A.A.A (eth0) IP : A.A.B.A (eth1) CPU : …. RAM : …. HDD : …. NIC : …. Victim (Server - Web) O/S : …. IP : A.A.B.B Client program : …. CPU : …. RAM : …. HDD : …. NIC : …. Sniffer - inbound O/S : …. IP : A.A.A.C Sniffer program : …. CPU : …. RAM : …. HDD : …. NIC : …. Sniffer - outbound O/S : …. IP : A.A.B.C Sniffer program : …. CPU : …. RAM : …. HDD : …. NIC : ….
· To be completed after the test	
Actual result	
Audit result	

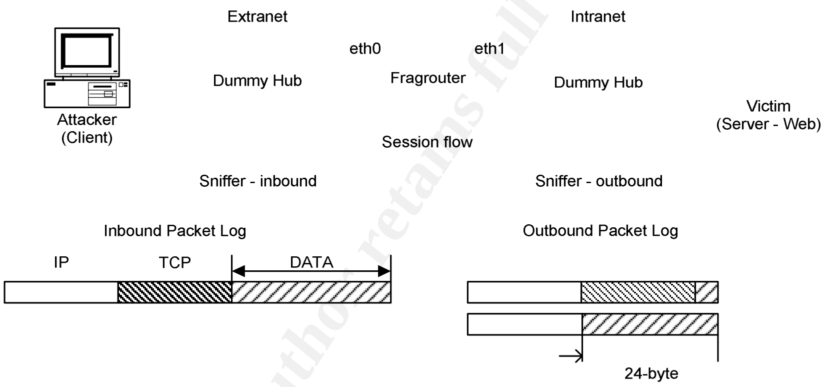
Item ID - 5	Baseline
Reference	･ “Insertion, Evasion, and Denial of Service: Eluding Network Intrusion Detection,” available via at http://www.securityfocus.com/data/library/ids.ps ･ Manpage of Fragrouter (*root#man fragrouter*) ･ See Sections 1.4 for complete listing of all references
Control objective	The objective of this item is to determine whether the Fragrouter can send the original data without any modification from inbound (client) interface to outbound (server) after a TCP handshake was completed. In other words, this functional testing performed by the tester establishes that the Fragrouter exhibits the properties necessary to analyze the vulnerability of NIDS.
Risk	If a failure of function happened, tester can not derive a completed and correct result from the test. Therefore, vulnerability test result to the NIDS can not be reliable and asset (information or resources) to be protected by NIDS may be exposed.
Compliance	･ The Fragrouter should perform as below followings. - Complete a TCP handshake - Send the test string in a single TCP data segment
Testing	･ step 1. Run sniffer - inbound & sniffer - outbound (capture mode) ･ step 2. Run Fragrouter (baseline) ･ step 3. Attempt to web connection from client to server ･ step 4. Confirm whether connection between client and server was established. If not connected, stop all programs and procedures, then try again from step 1 repeatedly ･ step 5. Stop sniffer - inbound & sniffer - outbound ･ step 6. Compare session log between sniffer - inbound and sniffer -

	outbound - Packets were transmitted as stated above through Fragrouter ? - If not as intended, think what is the wrong about testing procedures
Objective / Subjective	Objective – Results are generated repeatedly.
Expected result	 Extranet Intranet eth0 eth1 Dummy Hub Fragrouter Dummy Hub Victim (Server - Web) Attacker (Client) Session flow Sniffer - inbound Sniffer - outbound Inbound Packet Log Outbound Packet Log IP TCP DATA
· ***To be completed after the test***	
Actual result	
Audit result	

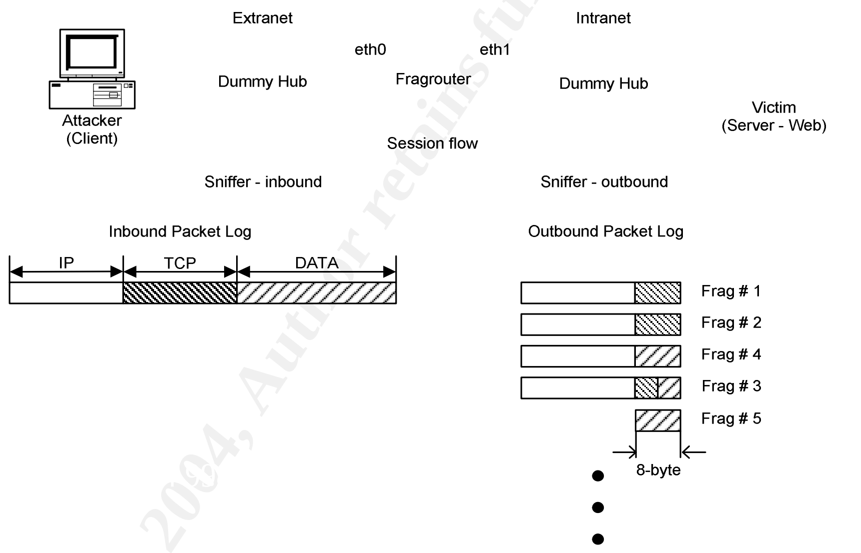
Item ID - 6	Frag - 1
Reference	· "Insertion, Evasion, and Denial of Service: Eluding Network Intrusion Detection," available via at http://www.securityfocus.com/data/library/ids.ps · Manpage of Fragrouter (*root#man fragrouter*) · See Sections 1.4 for complete listing of all references
Control objective	The objective of this item is to determine whether the Fragrouter can send data in ordered 8-byte IP fragments from inbound (client) interface to outbound (server) after a TCP handshake was completed. In other words, this functional testing performed by the tester establishes that the Fragrouter exhibits the properties necessary to analyze the vulnerability of NIDS.
Risk	If a failure of function happened, tester can not derive a completed and correct result from the test. Therefore, vulnerability test result to the NIDS can not be reliable and asset (information or resources) to be protected by NIDS may be exposed.
Compliance	· The Fragrouter should perform as below followings. - Complete a TCP handshake - Send the test string in a single TCP data segment which is broken into 8-byte IP fragments and sent in order
Testing	· step 1. Run sniffer - inbound & sniffer - outbound (capture mode) · step 2. Run Fragrouter (frag-1) · step 3. Attempt to web connection from client to server · step 4. Confirm whether connection between client and server was established. If not connected, stop all programs and procedures, then try again from step 1 repeatedly · step 5. Stop sniffer - inbound & sniffer - outbound · step 6.

	Compare session log between sniffer - inbound and sniffer - outbound - Packets were transmitted as stated above through Fragrouter ? - If not as intended, think what is the wrong about testing procedures
Objective / Subjective	Objective – Results are generated repeatedly.
Expected result	 Extranet Intranet eth0 eth1 Dummy Hub Fragrouter Dummy Hub Attacker (Client) Victim (Server - Web) Session flow Sniffer - inbound Sniffer - outbound Inbound Packet Log Outbound Packet Log IP TCP DATA 8-byte
· To be completed after the test	
Actual result	
Audit result	

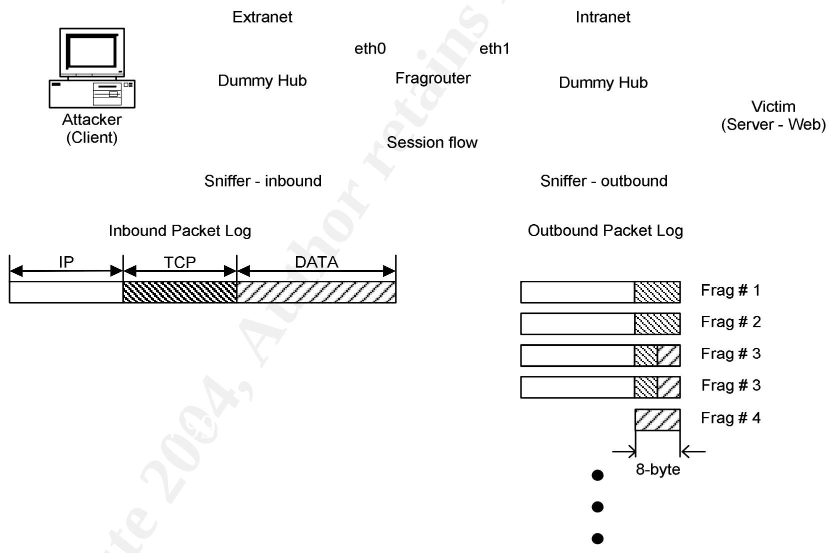
Item ID - 7	Frag - 2
Reference	· "Insertion, Evasion, and Denial of Service: Eluding Network Intrusion Detection," available via at http://www.securityfocus.com/data/library/ids.ps · Manpage of Fragrouter (*root#man fragrouter*) · See Sections 1.4 for complete listing of all references
Control objective	The objective of this item is to determine whether the Fragrouter can send data in ordered 24-byte IP fragments from inbound (client) interface to outbound (server) after a TCP handshake was completed. In other words, this functional testing performed by the tester establishes that the Fragrouter exhibits the properties necessary to analyze the vulnerability of NIDS.
Risk	If a failure of function happened, tester can not derive a completed and correct result from the test. Therefore, vulnerability test result to the NIDS can not be reliable and asset (information or resources) to be protected by NIDS may be exposed.
Compliance	· The Fragrouter should perform as below followings. - Complete a TCP handshake - Send the test string in a single TCP data segment which is broken into 24-byte IP fragments and sent in order
Testing	· step 1. Run sniffer - inbound & sniffer - outbound (capture mode) · step 2. Run Fragrouter (frag-2) · step 3. Attempt to web connection from client to server · step 4. Confirm whether connection between client and server was established. If not connected, stop all programs and procedures, then try again from step 1 repeatedly · step 5. Stop sniffer - inbound & sniffer - outbound · step 6.

	Compare session log between sniffer - inbound and sniffer - outbound - Packets were transmitted as stated above through Fragrouter ? - If not as intended, think what is the wrong about testing procedures
Objective / Subjective	Objective – Results are generated repeatedly.
Expected result	 Extranet Intranet eth0 eth1 Dummy Hub Fragrouter Dummy Hub Victim (Server - Web) Attacker (Client) Session flow Sniffer - inbound Sniffer - outbound Inbound Packet Log Outbound Packet Log IP TCP DATA 24-byte
· *To be completed after the test*	
Actual result	
Audit result	

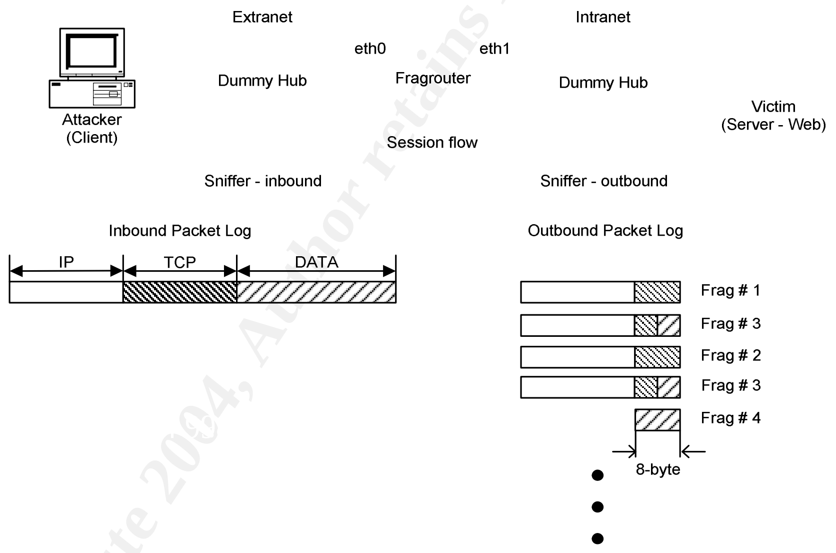
Item ID - 8	Frag - 3
Reference	▪ "Insertion, Evasion, and Denial of Service: Eluding Network Intrusion Detection," available via at http://www.securityfocus.com/data/library/ids.ps ▪ Manpage of Fragrouter (*root#man fragrouter*) ▪ See Sections 1.4 for complete listing of all references
Control objective	The objective of this item is to determine whether the Fragrouter can send data in ordered 8-byte IP fragments, with one fragment sent out of order from inbound (client) interface to outbound (server) after a TCP handshake was completed. In other words, this functional testing performed by the tester establishes that the Fragrouter exhibits the properties necessary to analyze the vulnerability of NIDS.
Risk	If a failure of function happened, tester can not derive a completed and correct result from the test. Therefore, vulnerability test result to the NIDS can not be reliable and asset (information or resources) to be protected by NIDS may be exposed.
Compliance	▪ The Fragrouter should perform as below followings. - Complete a TCP handshake - Send the test string in a single TCP data segment which is broken into 8-byte IP fragments, with one of those fragments sent out of order
Testing	▪ step 1. Run sniffer - inbound & sniffer - outbound (capture mode) ▪ step 2. Run Fragrouter (frag-3) ▪ step 3. Attempt to web connection from client to server ▪ step 4. Confirm whether connection between client and server was established. If not connected, stop all programs and procedures, then try again from step 1 repeatedly ▪ step 5. Stop sniffer - inbound & sniffer - outbound

	· step 6. Compare session log between sniffer - inbound and sniffer - outbound - Packets were transmitted as stated above through Fragrouter ? - If not as intended, think what is the wrong about testing procedures
Objective / Subjective	Objective – Results are generated repeatedly.
Expected result	
· *To be completed after the test*	
Actual result	
Audit result	

Item ID - 9	Frag - 4
Reference	･ "Insertion, Evasion, and Denial of Service: Eluding Network Intrusion Detection," available via at http://www.securityfocus.com/data/library/ids.ps ･ Manpage of Fragrouter (*root#man fragrouter*) ･ See Sections 1.4 for complete listing of all references
Control objective	The objective of this item is to determine whether the Fragrouter can send data in ordered 8-byte IP fragments, duplicating the penultimate fragment in each packet from inbound (client) interface to outbound (server) after a TCP handshake was completed. In other words, this functional testing performed by the tester establishes that the Fragrouter exhibits the properties necessary to analyze the vulnerability of NIDS.
Risk	If a failure of function happened, tester can not derive a completed and correct result from the test. Therefore, vulnerability test result to the NIDS can not be reliable and asset (information or resources) to be protected by NIDS may be exposed.
Compliance	･ The Fragrouter should perform as below followings. - Complete a TCP handshake - Send the test string in a single TCP data segment which is broken into 8-byte IP fragments, with one of those fragments sent twice
Testing	･ step 1. Run sniffer - inbound & sniffer - outbound (capture mode) ･ step 2. Run Fragrouter (frag-4) ･ step 3. Attempt to web connection from client to server ･ step 4. Confirm whether connection between client and server was established. If not connected, stop all programs and procedures, then try again from step 1 repeatedly ･ step 5.

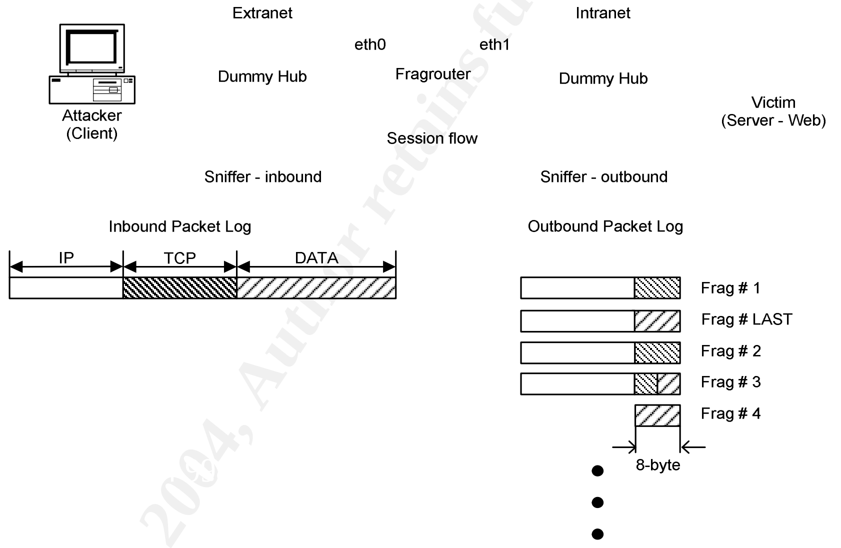
	Stop sniffer - inbound & sniffer - outbound · step 6. Compare session log between sniffer - inbound and sniffer - outbound - Packets were transmitted as stated above through Fragrouter ? - If not as intended, think what is the wrong about testing procedures
Objective / Subjective	Objective – Results are generated repeatedly.
Expected result	
· To be completed after the test	
Actual result	
Audit result	

Item ID - 10	Frag - 5
Reference	· "Insertion, Evasion, and Denial of Service: Eluding Network Intrusion Detection," available via at http://www.securityfocus.com/data/library/ids.ps · Manpage of Fragrouter (*root#man fragrouter*) · See Sections 1.4 for complete listing of all references
Control objective	The objective of this item is to determine whether the Fragrouter can send data in out of ordered 8-byte IP fragments, duplicating the penultimate fragment in each packet from inbound (client) interface to outbound (server) after a TCP handshake was completed. In other words, this functional testing performed by the tester establishes that the Fragrouter exhibits the properties necessary to analyze the vulnerability of NIDS.
Risk	If a failure of function happened, tester can not derive a completed and correct result from the test. Therefore, vulnerability test result to the NIDS can not be reliable and asset (information or resources) to be protected by NIDS may be exposed.
Compliance	· The Fragrouter should perform as below followings. - Complete a TCP handshake - Send the test string in a single TCP data segment which is broken into 8-byte IP fragments, sent completely out of order and with an arbitrary duplicated fragment.
Testing	· step 1. Run sniffer - inbound & sniffer - outbound (capture mode) · step 2. Run Fragrouter (frag-5) · step 3. Attempt to web connection from client to server · step 4. Confirm whether connection between client and server was established. If not connected, stop all programs and procedures, then try again from step 1 repeatedly · step 5.

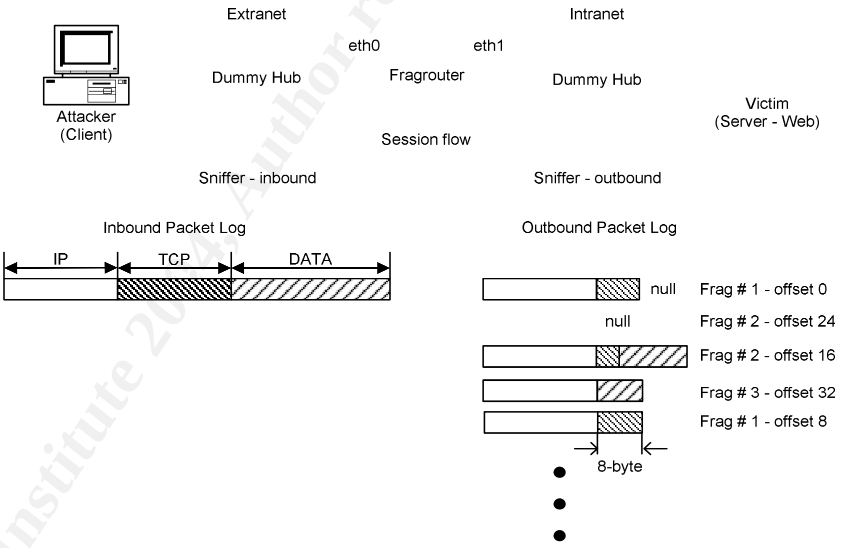
<table>
<tr><td></td><td>Stop sniffer - inbound & sniffer - outbound
· step 6.
Compare session log between sniffer - inbound and sniffer - outbound
- Packets were transmitted as stated above through Fragrouter ?
- If not as intended, think what is the wrong about testing procedures</td></tr>
<tr><td><b>Objective / Subjective</b></td><td>Objective – Results are generated repeatedly.</td></tr>
<tr><td><b>Expected result</b></td><td>


</td></tr>
<tr><td colspan="2"><b><i>· To be completed after the test</i></b></td></tr>
<tr><td><b>Actual result</b></td><td></td></tr>
<tr><td><b>Audit result</b></td><td></td></tr>
</table>

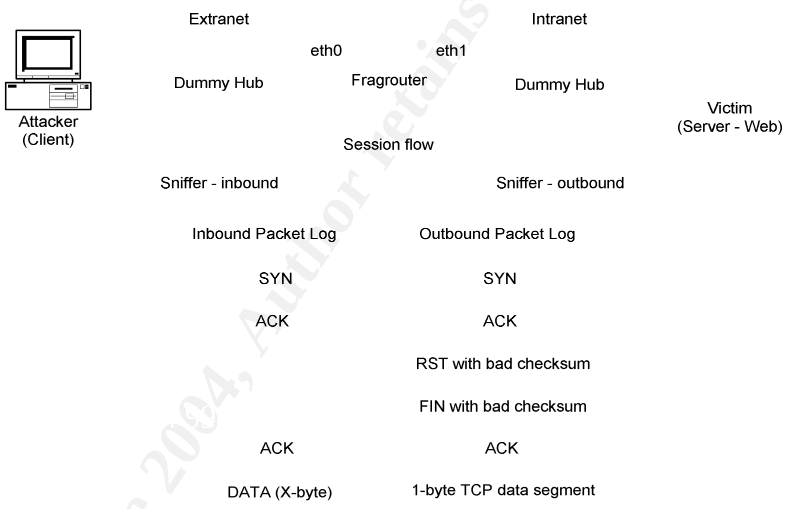
Item ID - 11	Frag - 6
Reference	· "Insertion, Evasion, and Denial of Service: Eluding Network Intrusion Detection," available via at http://www.securityfocus.com/data/library/ids.ps · Manpage of Fragrouter (*root#man fragrouter*) · See Sections 1.4 for complete listing of all references
Control objective	The objective of this item is to determine whether the Fragrouter can send data in ordered 8-byte IP fragments, sending the marked last fragment first from inbound (client) interface to outbound (server) after a TCP handshake was completed. In other words, this functional testing performed by the tester establishes that the Fragrouter exhibits the properties necessary to analyze the vulnerability of NIDS.
Risk	If a failure of function happened, tester can not derive a completed and correct result from the test. Therefore, vulnerability test result to the NIDS can not be reliable and asset (information or resources) to be protected by NIDS may be exposed.
Compliance	· The Fragrouter should perform as below followings. - Complete a TCP handshake - Send the test string in a single TCP data segment which is broken into 8-byte IP fragments, sending the marked last fragment before any of the others.
Testing	· step 1. Run sniffer - inbound & sniffer - outbound (capture mode) · step 2. Run Fragrouter (frag-6) · step 3. Attempt to web connection from client to server · step 4. Confirm whether connection between client and server was established. If not connected, stop all programs and procedures, then try again from step 1 repeatedly · step 5. Stop sniffer - inbound & sniffer - outbound

	· step 6. Compare session log between sniffer - inbound and sniffer - outbound - Packets were transmitted as stated above through Fragrouter ? - If not as intended, think what is the wrong about testing procedures
Objective / Subjective	Objective – Results are generated repeatedly.
Expected result	 Extranet Intranet eth0 eth1 Dummy Hub Fragrouter Dummy Hub Attacker (Client) Victim (Server - Web) Session flow Sniffer - inbound Sniffer - outbound Inbound Packet Log Outbound Packet Log IP TCP DATA Frag # 1 Frag # LAST Frag # 2 Frag # 3 Frag # 4 8-byte
· To be completed after the test	
Actual result	
Audit result	

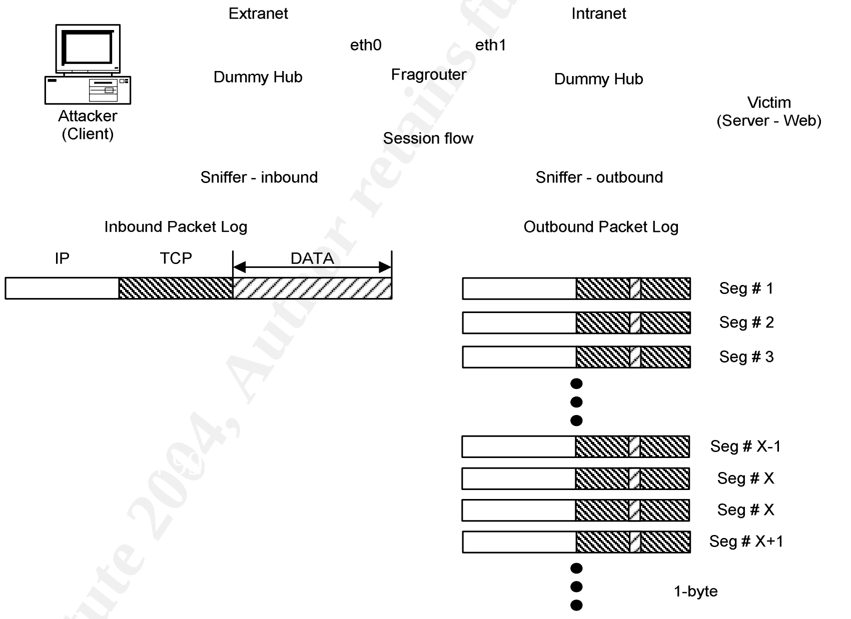
Item ID - 12	Frag - 7
Reference	· “Insertion, Evasion, and Denial of Service: Eluding Network Intrusion Detection,” available via at http://www.securityfocus.com/data/library/ids.ps · Manpage of Fragrouter (*root#man fragrouter*) · See Sections 1.4 for complete listing of all references
Control objective	The objective of this item is to determine whether the Fragrouter can send data in ordered 16-byte IP fragments, preceding each fragment with an 8-byte null data fragment that overlaps the latter half of it from inbound (client) interface to outbound (server) after a TCP handshake was completed. This amounts to the forward-overlapping 16-byte fragment rewriting the null data back to the real attack. In other words, this functional testing performed by the tester establishes that the Fragrouter exhibits the properties necessary to analyze the vulnerability of NIDS.
Risk	If a failure of function happened, tester can not derive a completed and correct result from the test. Therefore, vulnerability test result to the NIDS can not be reliable and asset (information or resources) to be protected by NIDS may be exposed.
Compliance	· The Fragrouter should perform as below followings. - Complete a TCP handshake - For examples, Send a stream of fragments containing the signature string with the word “GET” replaced with the string “SNI”. Send a forward-overlapping fragment rewriting the “SNI” back to “GET” on the target host.
Testing	· step 1. Run sniffer - inbound & sniffer - outbound (capture mode) · step 2. Run Fragrouter (frag-7) · step 3. Attempt to web connection from client to server · step 4. Confirm whether connection between client and server was

	established. If not connected, stop all programs and procedures, then try again from step 1 repeatedly · step 5. Stop sniffer - inbound & sniffer - outbound · step 6. Compare session log between sniffer - inbound and sniffer -outbound - Packets were transmitted as stated above through Fragrouter ? - If not as intended, think what is the wrong about testing procedures
Objective / Subjective	Objective – Results are generated repeatedly.
Expected result	 Extranet Intranet eth0 eth1 Dummy Hub Fragrouter Dummy Hub Victim (Server - Web) Attacker (Client) Session flow Sniffer - inbound Sniffer - outbound Inbound Packet Log Outbound Packet Log IP TCP DATA null Frag # 1 - offset 0 null Frag # 2 - offset 24 Frag # 2 - offset 16 Frag # 3 - offset 32 Frag # 1 - offset 8 8-byte
· ***To be completed after the test***	
Actual result	
Audit result	

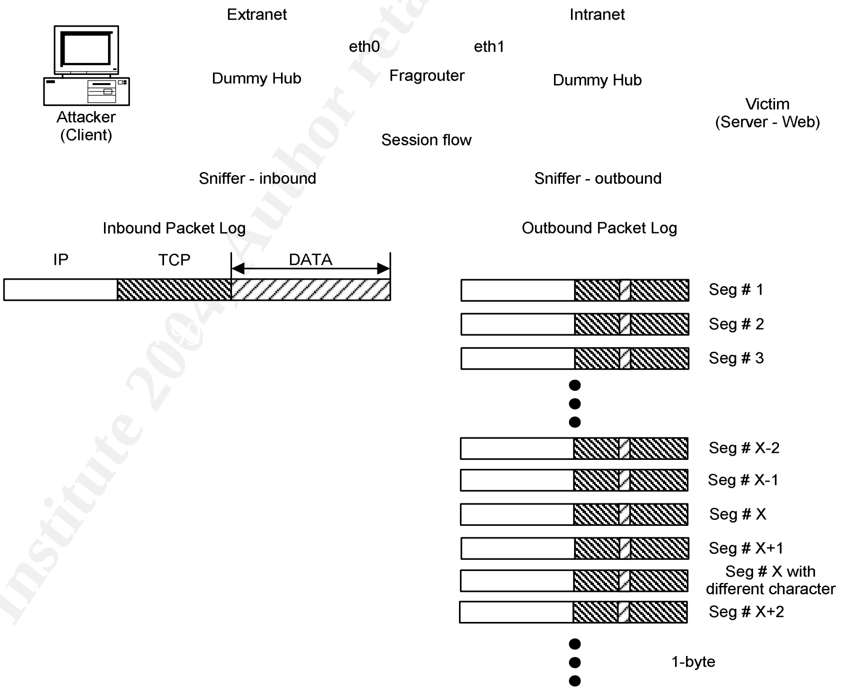
Item ID - 13	**TCP - 1**
Reference	· “Insertion, Evasion, and Denial of Service: Eluding Network Intrusion Detection,” available via at http://www.securityfocus.com/data/library/ids.ps · Manpage of Fragrouter (*root#man fragrouter*) · See Sections 1.4 for complete listing of all references
Control objective	The objective of this item is to determine whether the Fragrouter can send fake FIN and RST (with bad checksums) before sending data in ordered 1-byte segments from inbound (client) interface to outbound (server) after a TCP handshake was completed. In other words, this functional testing performed by the tester establishes that the Fragrouter exhibits the properties necessary to analyze the vulnerability of NIDS.
Risk	If a failure of function happened, tester can not derive a completed and correct result from the test. Therefore, vulnerability test result to the NIDS can not be reliable and asset (information or resources) to be protected by NIDS may be exposed.
Compliance	· The Fragrouter should perform as below followings. - Complete a TCP handshake - Simulate the disconnection of the target host from the network, and send the target string in a series of 1-byte TCP data segments.
Testing	· step 1. Run sniffer - inbound & sniffer - outbound (capture mode) · step 2. Run Fragrouter (tcp-1) · step 3. Attempt to web connection from client to server · step 4. Confirm whether connection between client and server was established. If not connected, stop all programs and procedures, then try again from step 1 repeatedly · step 5.

	Stop sniffer - inbound & sniffer - outbound · step 6. Compare session log between sniffer - inbound and sniffer -outbound - Packets were transmitted as stated above through Fragrouter ? - If not as intended, think what is the wrong about testing procedures
Objective / Subjective	Objective – Results are generated repeatedly.
Expected result	 Extranet Intranet eth0 eth1 Dummy Hub Fragrouter Dummy Hub Attacker (Client) Victim (Server - Web) Session flow Sniffer - inbound Sniffer - outbound Inbound Packet Log Outbound Packet Log SYN SYN ACK ACK RST with bad checksum FIN with bad checksum ACK ACK DATA (X-byte) 1-byte TCP data segment
· To be completed after the test	
Actual result	
Audit result	

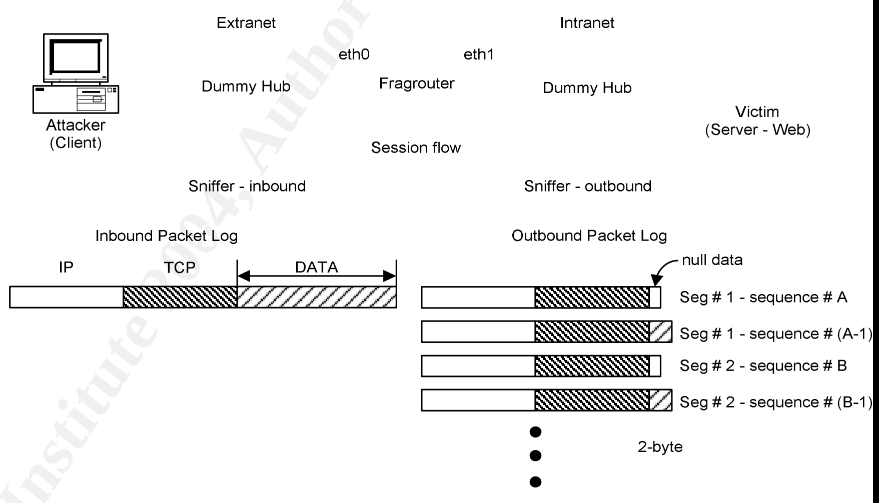
Item ID - 14	TCP - 3
Reference	・ "Insertion, Evasion, and Denial of Service: Eluding Network Intrusion Detection," available via at http://www.securityfocus.com/data/library/ids.ps ・ Manpage of Fragrouter (*root#man fragrouter*) ・ See Sections 1.4 for complete listing of all references
Control objective	The objective of this item is to determine whether the Fragrouter can send data in ordered 1-byte segments, duplicating the penultimate segment of each original TCP packet from inbound (client) interface to outbound (server) after a TCP handshake was completed. In other words, this functional testing performed by the tester establishes that the Fragrouter exhibits the properties necessary to analyze the vulnerability of NIDS.
Risk	If a failure of function happened, tester can not derive a completed and correct result from the test. Therefore, vulnerability test result to the NIDS can not be reliable and asset (information or resources) to be protected by NIDS may be exposed.
Compliance	・ The Fragrouter should perform as below followings. - Complete a TCP handshake - Send the test string in a stream of 1-byte TCP data segments, duplicating entirely one of those segments.
Testing	・ step 1. Run sniffer - inbound & sniffer - outbound (capture mode) ・ step 2. Run Fragrouter (tcp-3) ・ step 3. Attempt to web connection from client to server ・ step 4. Confirm whether connection between client and server was established. If not connected, stop all programs and procedures, then try again from step 1 repeatedly ・ step 5. Stop sniffer - inbound & sniffer - outbound

	· step 6. Compare session log between sniffer - inbound and sniffer -outbound - Packets were transmitted as stated above through Fragrouter ? - If not as intended, think what is the wrong about testing procedures
Objective / Subjective	Objective – Results are generated repeatedly.
Expected result	 Extranet Intranet eth0 eth1 Dummy Hub Fragrouter Dummy Hub Attacker (Client) Victim (Server - Web) Session flow Sniffer - inbound Sniffer - outbound Inbound Packet Log Outbound Packet Log IP TCP DATA Seg # 1 Seg # 2 Seg # 3 Seg # X-1 Seg # X Seg # X Seg # X+1 1-byte
· ***To be completed after the test***	
Actual result	
Audit result	

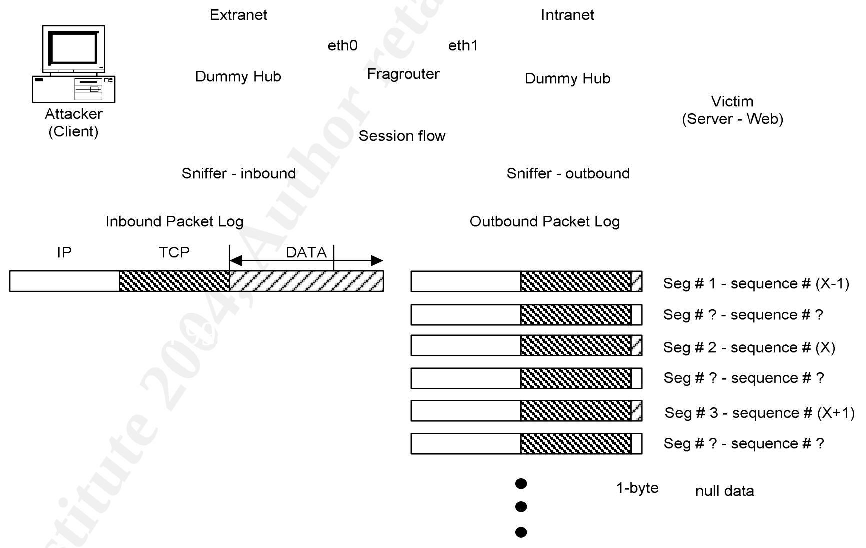
Item ID - 15	TCP - 4
Reference	▪ "Insertion, Evasion, and Denial of Service: Eluding Network Intrusion Detection," available via at http://www.securityfocus.com/data/library/ids.ps ▪ Manpage of Fragrouter (*root#man fragrouter*) ▪ See Sections 1.4 for complete listing of all references
Control objective	The objective of this item is to determine whether the Fragrouter can send data in ordered 1-byte segments, sending additional 1-byte segment which overlaps the penultimate segment of each original TCP packet with a null data payload from inbound (client) interface to outbound (server) after a TCP handshake was completed. In other words, this functional testing performed by the tester establishes that the Fragrouter exhibits the properties necessary to analyze the vulnerability of NIDS.
Risk	If a failure of function happened, tester can not derive a completed and correct result from the test. Therefore, vulnerability test result to the NIDS can not be reliable and asset (information or resources) to be protected by NIDS may be exposed.
Compliance	▪ The Fragrouter should perform as below followings. - Complete a TCP handshake - Send the test string in a stream of 1-byte TCP data segments, sending an additional 1-byte TCP segment which overlaps a previous segment completely but contains a different character.
Testing	▪ step 1. Run sniffer - inbound & sniffer - outbound (capture mode) ▪ step 2. Run Fragrouter (tcp-4) ▪ step 3. Attempt to web connection from client to server ▪ step 4. Confirm whether connection between client and server was established. If not connected, stop all programs and

	procedures, then try again from step 1 repeatedly · step 5. Stop sniffer - inbound & sniffer - outbound · step 6. Compare session log between sniffer - inbound and sniffer -outbound - Packets were transmitted as stated above through Fragrouter ? - If not as intended, think what is the wrong about testing procedures
Objective / Subjective	Objective – Results are generated repeatedly.
Expected result	 Extranet Intranet eth0 eth1 Dummy Hub Fragrouter Dummy Hub Victim (Server - Web) Attacker (Client) Session flow Sniffer - inbound Sniffer - outbound Inbound Packet Log Outbound Packet Log IP TCP DATA Seg # 1 Seg # 2 Seg # 3 Seg # X-2 Seg # X-1 Seg # X Seg # X+1 Seg # X with different character Seg # X+2 1-byte
· To be completed after the test	
Actual result	
Audit result	

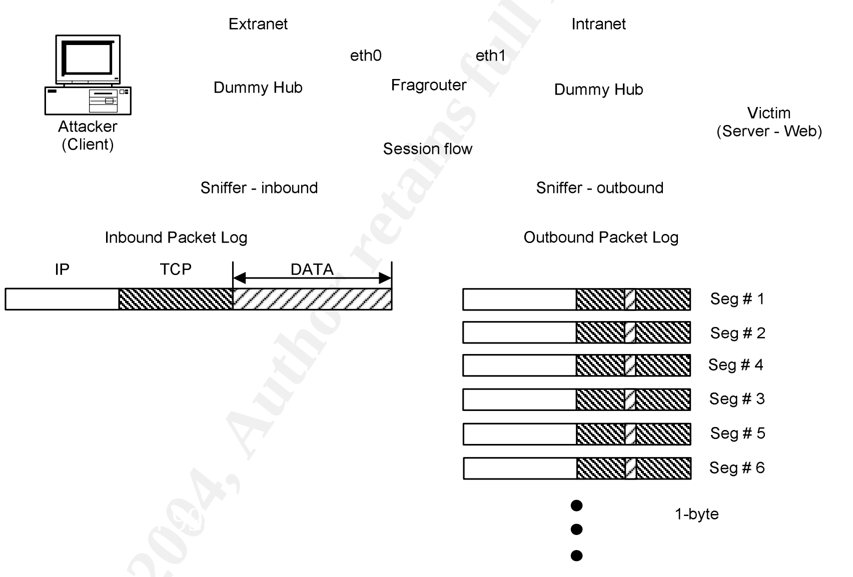
Item ID - 16	TCP - 5
Reference	･ "Insertion, Evasion, and Denial of Service: Eluding Network Intrusion Detection," available via at http://www.securityfocus.com/data/library/ids.ps ･ Manpage of Fragrouter (*root#man fragrouter*) ･ See Sections 1.4 for complete listing of all references
Control objective	The objective of this item is to determine whether the Fragrouter can send data in ordered 2-byte segments, preceding each segment with a 1-byte null data segment that overlaps the latter half of it from inbound (client) interface to outbound (server) after a TCP handshake was completed. This amounts to the forward-overlapping 2-byte segment rewriting the null data back to the real attack. In other words, this functional testing performed by the tester establishes that the Fragrouter exhibits the properties necessary to analyze the vulnerability of NIDS.
Risk	If a failure of function happened, tester can not derive a completed and correct result from the test. Therefore, vulnerability test result to the NIDS can not be reliable and asset (information or resources) to be protected by NIDS may be exposed.
Compliance	･ The Fragrouter should perform as below followings. - Complete a TCP handshake - For examples, send the test string, with the letter "c" replaced with the letter "X", in a series of 1-byte TCP data segments. Immediately send a 2-byte TCP data segment that overlaps (forward) the modified letter, rewriting it back to "c" on the target host.
Testing	･ step 1. Run sniffer - inbound & sniffer - outbound (capture mode) ･ step 2. Run Fragrouter (tcp-5) ･ step 3. Attempt to web connection from client to server ･ step 4.

	Confirm whether connection between client and server was established. If not connected, stop all programs and procedures, then try again from step 1 repeatedly · step 5. Stop sniffer - inbound & sniffer - outbound · step 6. Compare session log between sniffer - inbound and sniffer -outbound - Packets were transmitted as stated above through Fragrouter ? - If not as intended, think what is the wrong about testing procedures
Objective / Subjective	Objective – Results are generated repeatedly.
Expected result	 Extranet Intranet eth0 eth1 Dummy Hub Fragrouter Dummy Hub Attacker (Client) Victim (Server - Web) Session flow Sniffer - inbound Sniffer - outbound Inbound Packet Log Outbound Packet Log IP TCP DATA null data Seg # 1 - sequence # A Seg # 1 - sequence # (A-1) Seg # 2 - sequence # B Seg # 2 - sequence # (B-1) 2-byte
· ***To be completed after the test***	
Actual result	
Audit result	

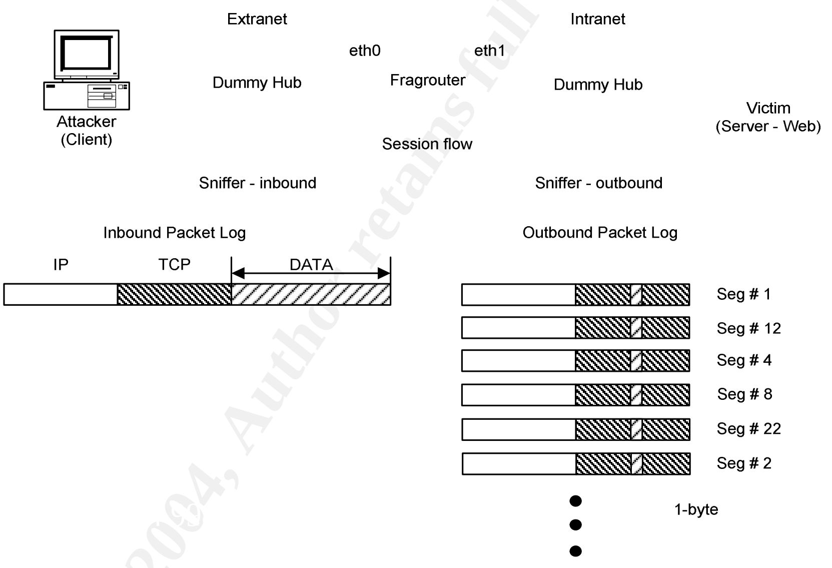
Item ID - 17	**TCP - 7**
Reference	▪ “Insertion, Evasion, and Denial of Service: Eluding Network Intrusion Detection,” available via at http://www.securityfocus.com/data/library/ids.ps ▪ Manpage of Fragrouter (*root#man fragrouter*) ▪ See Sections 1.4 for complete listing of all references
Control objective	The objective of this item is to determine whether the Fragrouter can send data in ordered 1-byte segments interleaved with 1-byte null segments for the same connection but with drastically different sequence numbers from inbound (client) interface to outbound (server) after a TCP handshake was completed. In other words, this functional testing performed by the tester establishes that the Fragrouter exhibits the properties necessary to analyze the vulnerability of NIDS.
Risk	If a failure of function happened, tester can not derive a completed and correct result from the test. Therefore, vulnerability test result to the NIDS can not be reliable and asset (information or resources) to be protected by NIDS may be exposed.
Compliance	▪ The Fragrouter should perform as below followings. - Complete a TCP handshake - Send the test string in a series of 1-byte TCP data segments, interleaved with a stream of 1-byte data segments for the same connection but with drastically different sequence numbers.
Testing	▪ step 1. Run sniffer - inbound & sniffer - outbound (capture mode) ▪ step 2. Run Fragrouter (tcp-7) ▪ step 3. Attempt to web connection from client to server ▪ step 4. Confirm whether connection between client and server was established. If not connected, stop all programs and

	procedures, then try again from step 1 repeatedly · step 5. Stop sniffer - inbound & sniffer - outbound · step 6. Compare session log between sniffer - inbound and sniffer -outbound - Packets were transmitted as stated above through Fragrouter ? - If not as intended, think what is the wrong about testing procedures
Objective / Subjective	Objective – Results are generated repeatedly.
Expected result	 Extranet Intranet eth0 eth1 Dummy Hub Fragrouter Dummy Hub Attacker (Client) Victim (Server - Web) Session flow Sniffer - inbound Sniffer - outbound Inbound Packet Log Outbound Packet Log IP TCP DATA Seg # 1 - sequence # (X-1) Seg # ? - sequence # ? Seg # 2 - sequence # (X) Seg # ? - sequence # ? Seg # 3 - sequence # (X+1) Seg # ? - sequence # ? 1-byte null data
· To be completed after the test	
Actual result	
Audit result	

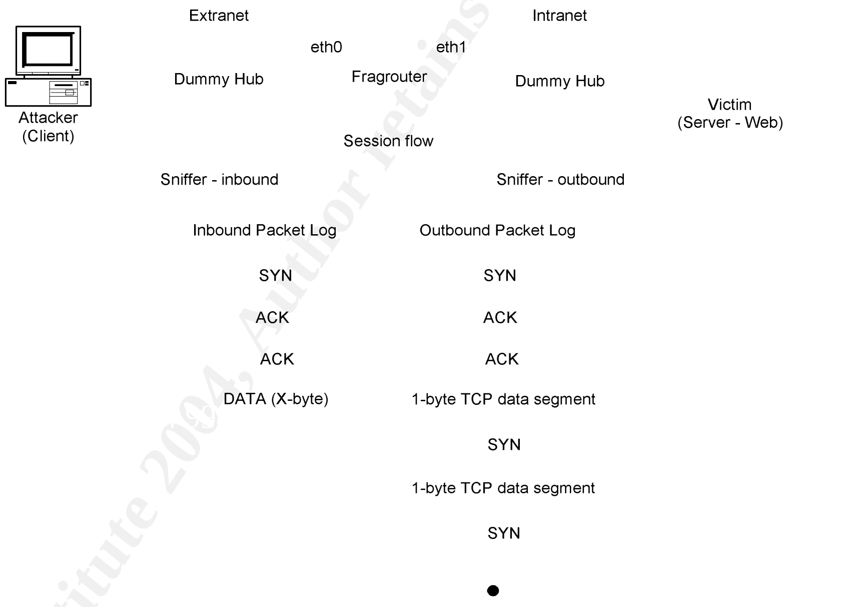
Item ID - 18	TCP - 8
Reference	▪ "Insertion, Evasion, and Denial of Service: Eluding Network Intrusion Detection," available via at http://www.securityfocus.com/data/library/ids.ps ▪ Manpage of Fragrouter (*root#man fragrouter*) ▪ See Sections 1.4 for complete listing of all references
Control objective	The objective of this item is to determine whether the Fragrouter can send data in ordered 1-byte segments with one segment send out of order from inbound (client) interface to outbound (server) after a TCP handshake was completed. In other words, this functional testing performed by the tester establishes that the Fragrouter exhibits the properties necessary to analyze the vulnerability of NIDS.
Risk	If a failure of function happened, tester can not derive a completed and correct result from the test. Therefore, vulnerability test result to the NIDS can not be reliable and asset (information or resources) to be protected by NIDS may be exposed.
Compliance	▪ The Fragrouter should perform as below followings. - Complete a TCP handshake - Send the test string in a series of 1-byte TCP data segments, with one of those segments sent out of order.
Testing	▪ step 1. Run sniffer - inbound & sniffer - outbound (capture mode) ▪ step 2. Run Fragrouter (tcp-8) ▪ step 3. Attempt to web connection from client to server ▪ step 4. Confirm whether connection between client and server was established. If not connected, stop all programs and procedures, then try again from step 1 repeatedly ▪ step 5. Stop sniffer - inbound & sniffer - outbound ▪ step 6.

	Compare session log between sniffer - inbound and sniffer -outbound - Packets were transmitted as stated above through Fragrouter ? - If not as intended, think what is the wrong about testing procedures
Objective / Subjective	Objective – Results are generated repeatedly.
Expected result	
· To be completed after the test	
Actual result	
Audit result	

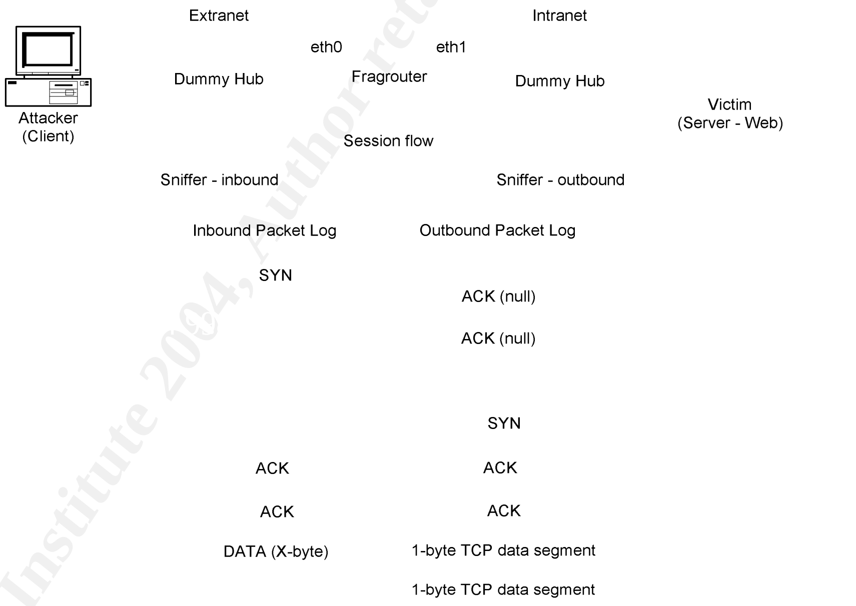
Item ID - 19	TCP - 9
Reference	· “Insertion, Evasion, and Denial of Service: Eluding Network Intrusion Detection,” available via at http://www.securityfocus.com/data/library/ids.ps · Manpage of Fragrouter (*root#man fragrouter*) · See Sections 1.4 for complete listing of all references
Control objective	The objective of this item is to determine whether the Fragrouter can send data in out of ordered 1-byte segments from inbound (client) interface to outbound (server) after a TCP handshake was completed. In other words, this functional testing performed by the tester establishes that the Fragrouter exhibits the properties necessary to analyze the vulnerability of NIDS.
Risk	If a failure of function happened, tester can not derive a completed and correct result from the test. Therefore, vulnerability test result to the NIDS can not be reliable and asset (information or resources) to be protected by NIDS may be exposed.
Compliance	· The Fragrouter should perform as below followings. - Complete a TCP handshake - Send the test string in a series of 1-byte TCP data segments, send in random order.
Testing	· step 1. Run sniffer - inbound & sniffer - outbound (capture mode) · step 2. Run Fragrouter (tcp-9) · step 3. Attempt to web connection from client to server · step 4. Confirm whether connection between client and server was established. If not connected, stop all programs and procedures, then try again from step 1 repeatedly · step 5. Stop sniffer - inbound & sniffer - outbound · step 6.

	Compare session log between sniffer - inbound and sniffer -outbound - Packets were transmitted as stated above through Fragrouter ? - If not as intended, think what is the wrong about testing procedures
Objective / Subjective	Objective – Results are generated repeatedly.
Expected result	 Extranet Intranet eth0 eth1 Dummy Hub Fragrouter Dummy Hub Attacker (Client) Victim (Server - Web) Session flow Sniffer - inbound Sniffer - outbound Inbound Packet Log Outbound Packet Log IP TCP DATA Seg # 1 Seg # 12 Seg # 4 Seg # 8 Seg # 22 Seg # 2 1-byte
· To be completed after the test	
Actual result	
Audit result	

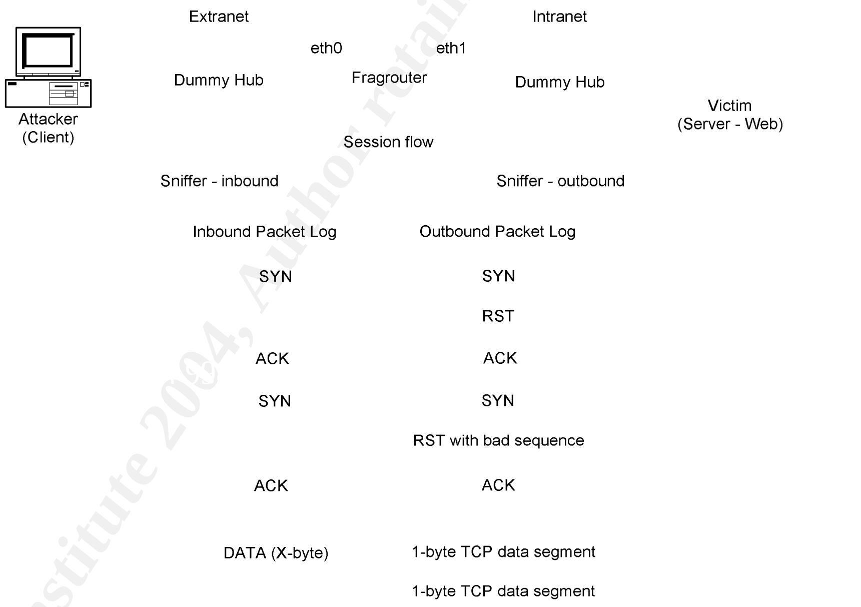
Item ID - 20	TCBC - 2
Reference	▪ "Insertion, Evasion, and Denial of Service: Eluding Network Intrusion Detection," available via at http://www.securityfocus.com/data/library/ids.ps ▪ Manpage of Fragrouter (*root#man fragrouter*) ▪ See Sections 1.4 for complete listing of all references
Control objective	The objective of this item is to determine whether the Fragrouter can send data in ordered 1-byte segments interleaved with SYN packets for the same connection parameters from inbound (client) interface to outbound (server) after a TCP handshake was completed. In other words, this functional testing performed by the tester establishes that the Fragrouter exhibits the properties necessary to analyze the vulnerability of NIDS.
Risk	If a failure of function happened, tester can not derive a completed and correct result from the test. Therefore, vulnerability test result to the NIDS can not be reliable and asset (information or resources) to be protected by NIDS may be exposed.
Compliance	▪ The Fragrouter should perform as below followings. - Complete a TCP handshake - Send the test string in a series of 1-byte TCP segments, interleaved with SYN packets for the same connection parameters.
Testing	▪ step 1. Run sniffer - inbound & sniffer - outbound (capture mode) ▪ step 2. Run Fragrouter (tcbc-2) ▪ step 3. Attempt to web connection from client to server ▪ step 4. Confirm whether connection between client and server was established. If not connected, stop all programs and procedures, then try again from step 1 repeatedly ▪ step 5.

	Stop sniffer - inbound & sniffer - outbound · step 6. Compare session log between sniffer - inbound and sniffer -outbound - Packets were transmitted as stated above through Fragrouter ? - If not as intended, think what is the wrong about testing procedures
Objective / Subjective	Objective – Results are generated repeatedly.
Expected result	 Extranet Intranet eth0 eth1 Dummy Hub Fragrouter Dummy Hub Attacker (Client) Victim (Server - Web) Session flow Sniffer - inbound Sniffer - outbound Inbound Packet Log Outbound Packet Log SYN SYN ACK ACK ACK ACK DATA (X-byte) 1-byte TCP data segment SYN 1-byte TCP data segment SYN ●
· To be completed after the test	
Actual result	
Audit result	

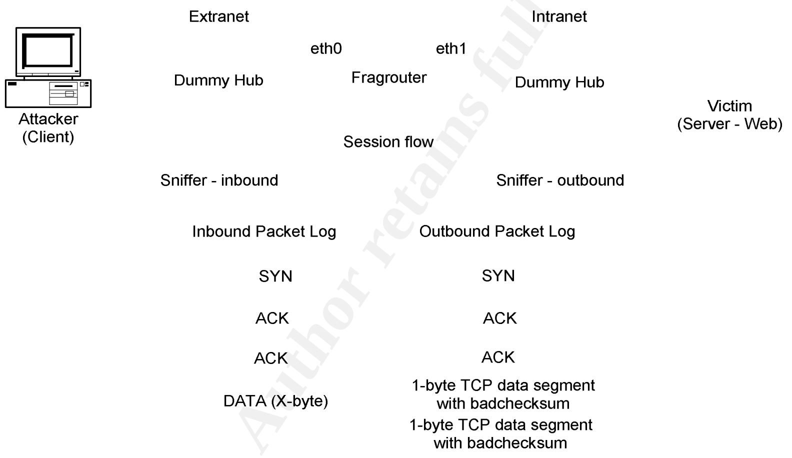
Item ID - 21	TCBC - 3
Reference	· “Insertion, Evasion, and Denial of Service: Eluding Network Intrusion Detection,” available via at http://www.securityfocus.com/data/library/ids.ps · Manpage of Fragrouter (*root#man fragrouter*) · See Sections 1.4 for complete listing of all references
Control objective	The objective of this item is to determine whether the Fragrouter can send null data in ordered 1-byte segments as if one had occurred from inbound (client) interface to outbound (server) before a TCP handshake was completed. Then, complete a TCP handshake with same connection parameters, and send the real data in ordered 1-byte segments. In other words, this functional testing performed by the tester establishes that the Fragrouter exhibits the properties necessary to analyze the vulnerability of NIDS.
Risk	If a failure of function happened, tester can not derive a completed and correct result from the test. Therefore, vulnerability test result to the NIDS can not be reliable and asset (information or resources) to be protected by NIDS may be exposed.
Compliance	· The Fragrouter should perform as below followings. - Do not complete a TCP handshake - But send a stream of arbitrary data at a random sequence number as if one had occurred. Use the same connection parameters to connect “netcat” and type the test string in manually.
Testing	· step 1. Run sniffer - inbound & sniffer - outbound (capture mode) · step 2. Run Fragrouter (tcbc-3) · step 3. Attempt to web connection from client to server · step 4. Confirm whether connection between client and server was established. If not connected, stop all programs and

	procedures, then try again from step 1 repeatedly · step 5. Stop sniffer - inbound & sniffer - outbound · step 6. Compare session log between sniffer - inbound and sniffer -outbound - Packets were transmitted as stated above through Fragrouter ? - If not as intended, think what is the wrong about testing procedures
Objective / Subjective	Objective – Results are generated repeatedly.
Expected result	
· To be completed after the test	
Actual result	
Audit result	

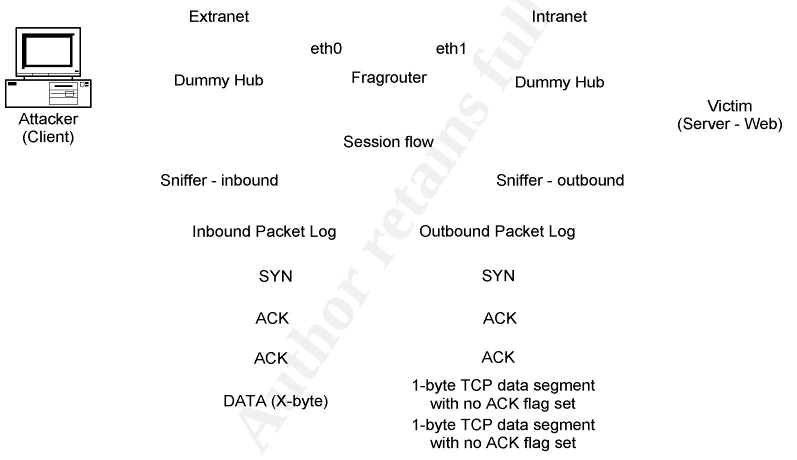
Item ID - 22	TCBT - 1
Reference	▪ “Insertion, Evasion, and Denial of Service: Eluding Network Intrusion Detection,” available via at http://www.securityfocus.com/data/library/ids.ps ▪ Manpage of Fragrouter (*root#man fragrouter*) ▪ See Sections 1.4 for complete listing of all references
Control objective	The objective of this item is to determine whether the Fragrouter can shut connection down with a RST, re-connect with drastically different sequence numbers and send data in ordered 1-byte segments from inbound (client) interface to outbound (server) after a TCP handshake was completed. In other words, this functional testing performed by the tester establishes that the Fragrouter exhibits the properties necessary to analyze the vulnerability of NIDS.
Risk	If a failure of function happened, tester can not derive a completed and correct result from the test. Therefore, vulnerability test result to the NIDS can not be reliable and asset (information or resources) to be protected by NIDS may be exposed.
Compliance	▪ The Fragrouter should perform as below followings. - Do not complete a TCP handshake - Immediately shut the connection down with an RST. Re-connect over the same parameters, with drastically different sequence numbers, and send the test string in a series of 1-byte TCP data segments.
Testing	▪ step 1. Run sniffer - inbound & sniffer - outbound (capture mode) ▪ step 2. Run Fragrouter (tcbt-1) ▪ step 3. Attempt to web connection from client to server ▪ step 4. Confirm whether connection between client and server was established. If not connected, stop all programs and procedures, then try again from step 1 repeatedly

	· step 5. Stop sniffer - inbound & sniffer - outbound · step 6. Compare session log between sniffer - inbound and sniffer -outbound - Packets were transmitted as stated above through Fragrouter ? - If not as intended, think what is the wrong about testing procedures
Objective / Subjective	Objective – Results are generated repeatedly.
Expected result	 Extranet Intranet eth0 eth1 Dummy Hub Fragrouter Dummy Hub Attacker (Client) Victim (Server - Web) Session flow Sniffer - inbound Sniffer - outbound Inbound Packet Log Outbound Packet Log SYN SYN RST ACK ACK SYN SYN RST with bad sequence ACK ACK DATA (X-byte) 1-byte TCP data segment 1-byte TCP data segment
· To be completed after the test	
Actual result	
Audit result	

Item ID - 23	INS - 2
Reference	▪ "Insertion, Evasion, and Denial of Service: Eluding Network Intrusion Detection," available via at http://www.securityfocus.com/data/library/ids.ps ▪ Manpage of Fragrouter (*root#man fragrouter*) ▪ See Sections 1.4 for complete listing of all references
Control objective	The objective of this item is to determine whether the Fragrouter can send data in ordered 1-byte segments but with bad TCP checksums from inbound (client) interface to outbound (server) after a TCP handshake was completed. In other words, this functional testing performed by the tester establishes that the Fragrouter exhibits the properties necessary to analyze the vulnerability of NIDS.
Risk	If a failure of function happened, tester can not derive a completed and correct result from the test. Therefore, vulnerability test result to the NIDS can not be reliable and asset (information or resources) to be protected by NIDS may be exposed.
Compliance	▪ The Fragrouter should perform as below followings. - Do not complete a TCP handshake - Send the test string in a series of 1-byte TCP data segments, each with a bad IP checksum.
Testing	▪ step 1. Run sniffer - inbound & sniffer - outbound (capture mode) ▪ step 2. Run Fragrouter (ins-2) ▪ step 3. Attempt to web connection from client to server ▪ step 4. Confirm whether connection between client and server was established. If not connected, stop all programs and procedures, then try again from step 1 repeatedly ▪ step 5. Stop sniffer - inbound & sniffer - outbound ▪ step 6.

	Compare session log between sniffer - inbound and sniffer -outbound - Packets were transmitted as stated above through Fragrouter ? - If not as intended, think what is the wrong about testing procedures
Objective / Subjective	Objective – Results are generated repeatedly.
Expected result	 Extranet Intranet eth0 eth1 Dummy Hub Fragrouter Dummy Hub Victim (Server - Web) Attacker (Client) Session flow Sniffer - inbound Sniffer - outbound Inbound Packet Log Outbound Packet Log SYN SYN ACK ACK ACK ACK DATA (X-byte) 1-byte TCP data segment with badchecksum 1-byte TCP data segment with badchecksum
· To be completed after the test	
Actual result	
Audit result	

Item ID - 24	INS - 3
Reference	▪ "Insertion, Evasion, and Denial of Service: Eluding Network Intrusion Detection," available via at http://www.securityfocus.com/data/library/ids.ps ▪ Manpage of Fragrouter (*root#man fragrouter*) ▪ See Sections 1.4 for complete listing of all references
Control objective	The objective of this item is to determine whether the Fragrouter can send data in ordered 1-byte segments but with no ACJ flag set from inbound (client) interface to outbound (server) after a TCP handshake was completed. In other words, this functional testing performed by the tester establishes that the Fragrouter exhibits the properties necessary to analyze the vulnerability of NIDS.
Risk	If a failure of function happened, tester can not derive a completed and correct result from the test. Therefore, vulnerability test result to the NIDS can not be reliable and asset (information or resources) to be protected by NIDS may be exposed.
Compliance	▪ The Fragrouter should perform as below followings. - Do not complete a TCP handshake - Send the test string in a series of 1-byte TCP data segments, none of which have the ACK bit set.
Testing	▪ step 1. Run sniffer - inbound & sniffer - outbound (capture mode) ▪ step 2. Run Fragrouter (ins-3) ▪ step 3. Attempt to web connection from client to server ▪ step 4. Confirm whether connection between client and server was established. If not connected, stop all programs and procedures, then try again from step 1 repeatedly ▪ step 5. Stop sniffer - inbound & sniffer - outbound ▪ step 6.

	Compare session log between sniffer - inbound and sniffer -outbound - Packets were transmitted as stated above through Fragrouter ? - If not as intended, think what is the wrong about testing procedures
Objective / Subjective	Objective – Results are generated repeatedly.
Expected result	 Extranet Intranet eth0 eth1 Dummy Hub Fragrouter Dummy Hub Victim (Server - Web) Attacker (Client) Session flow Sniffer - inbound Sniffer - outbound Inbound Packet Log Outbound Packet Log SYN SYN ACK ACK ACK ACK DATA (X-byte) 1-byte TCP data segment with no ACK flag set 1-byte TCP data segment with no ACK flag set
· *To be completed after the test*	
Actual result	
Audit result	

Assignment 3 – Audit Evidence

3.1– Conduct the Audit

The following 10 items have been chosen to from the above checklist and the results shown.

· Category : Environment

1. Item ID - 4 : Network configuration security

· Category : Functionality

2. Item ID - 5 : Baseline
3. Item ID - 6 : Frag – 1
4. Item ID - 8 : Frag – 3
5. Item ID - 10 : Frag - 5
6. Item ID - 13 : TCP - 1
7. Item ID - 16 : TCP - 5
8. Item ID - 20 : TCBC - 2
9. Item ID - 22 : TCBT - 1
10. Item ID - 24 : INS – 3

Item ID - 4	Network configuration security

▪ Testing

· step 1.

The auditor shall determine whether test environment is independent or not.

- Does Fragrouter take a role of gateway between two different networks ?
- Is there no any system that is unrelated to test environment ?

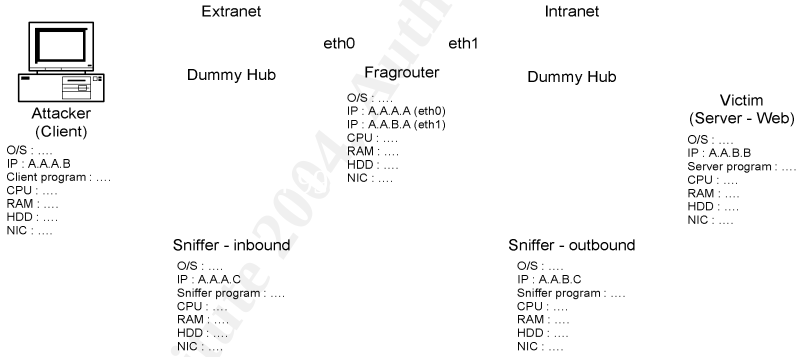
· step 2.

The auditor shall determine whether IP address is properly assigned to the system or not.

- Is assigned IP address valid ?
- Is there any duplication of IP address ?

▪ Expected result

Following is the preferred network environment configuration that is used for vulnerability analysis test on the NIDS.



Attacker (Client)
O/S : ….
IP : A.A.A.B
Client program : ….
CPU : ….
RAM : ….
HDD : ….
NIC : ….

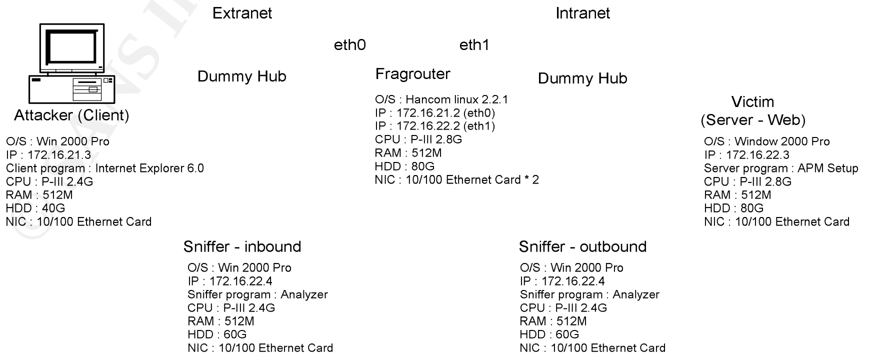
Fragrouter
O/S : ….
IP : A.A.A.A (eth0)
IP : A.A.B.A (eth1)
CPU : ….
RAM : ….
HDD : ….
NIC : ….

Victim (Server - Web)
O/S : ….
IP : A.A.B.B
Server program : ….
CPU : ….
RAM : ….
HDD : ….
NIC : ….

Sniffer - inbound
O/S : ….
IP : A.A.A.C
Sniffer program : ….
CPU : ….
RAM : ….
HDD : ….
NIC : ….

Sniffer - outbound
O/S : ….
IP : A.A.B.C
Sniffer program : ….
CPU : ….
RAM : ….
HDD : ….
NIC : ….

▪ Actual result



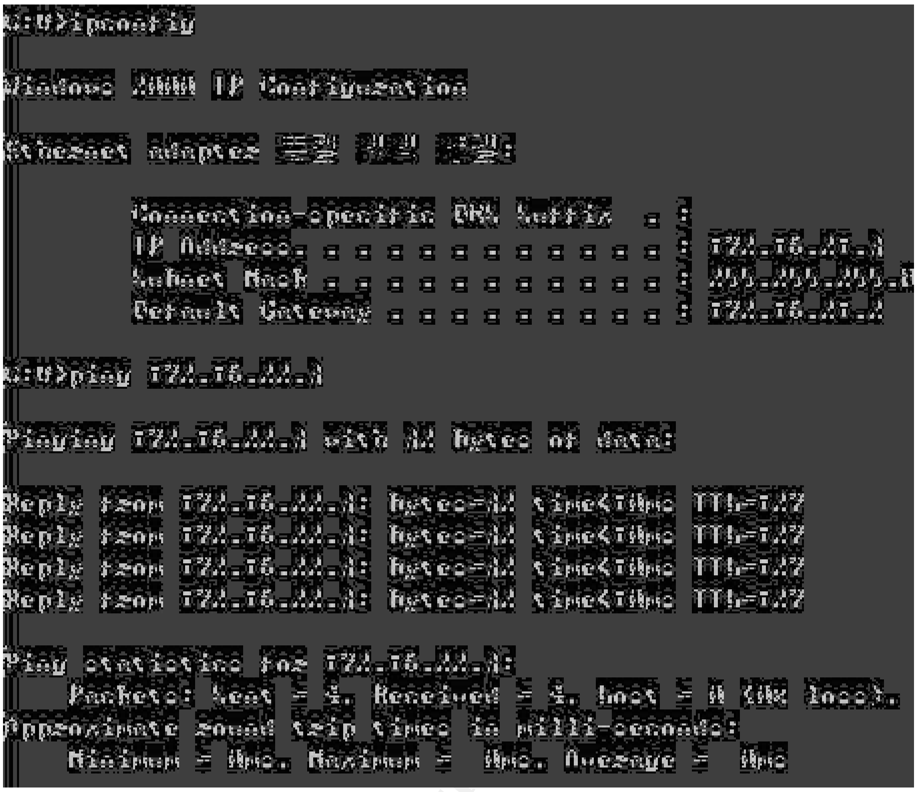
Attacker (Client)
O/S : Win 2000 Pro
IP : 172.16.21.3
Client program : Internet Explorer 6.0
CPU : P-III 2.4G
RAM : 512M
HDD : 40G
NIC : 10/100 Ethernet Card

Fragrouter
O/S : Hancom linux 2.2.1
IP : 172.16.21.2 (eth0)
IP : 172.16.22.2 (eth1)
CPU : P-III 2.8G
RAM : 512M
HDD : 80G
NIC : 10/100 Ethernet Card * 2

Victim (Server - Web)
O/S : Window 2000 Pro
IP : 172.16.22.3
Server program : APM Setup
CPU : P-III 2.8G
RAM : 512M
HDD : 80G
NIC : 10/100 Ethernet Card

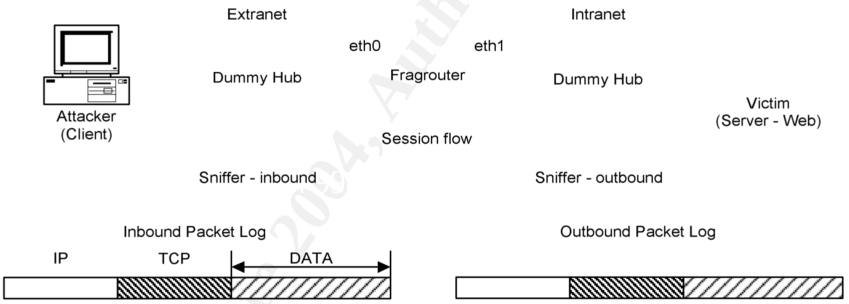
Sniffer - inbound
O/S : Win 2000 Pro
IP : 172.16.22.4
Sniffer program : Analyzer
CPU : P-III 2.4G
RAM : 512M
HDD : 60G
NIC : 10/100 Ethernet Card

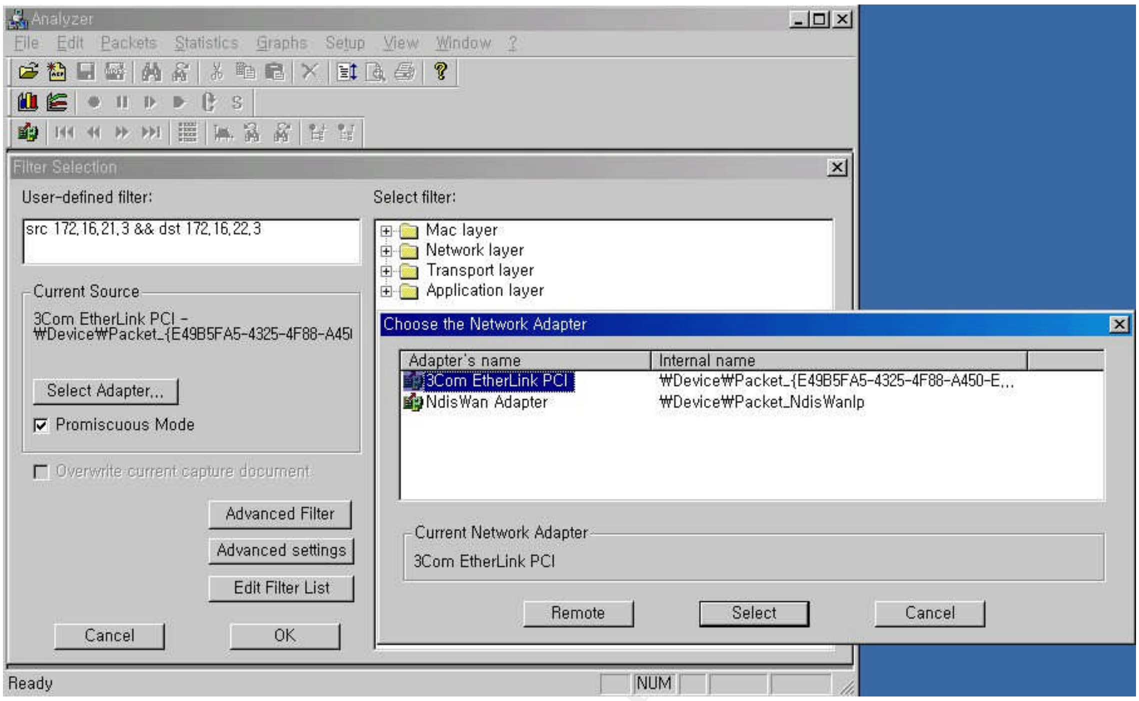
Sniffer - outbound
O/S : Win 2000 Pro
IP : 172.16.22.4
Sniffer program : Analyzer
CPU : P-III 2.4G
RAM : 512M
HDD : 60G
NIC : 10/100 Ethernet Card



```
C:\U>ipconfig

Windows 2000 IP Configuration

Ethernet adapter 로컬 영역 연결:

        Connection-specific DNS Suffix  . :
        IP Address. . . . . . . . . . . . : 172.16.21.1
        Subnet Mask . . . . . . . . . . . : 255.255.255.0
        Default Gateway . . . . . . . . . : 172.16.21.2

C:\U>ping 172.16.22.1

Pinging 172.16.22.1 with 32 bytes of data:

Reply from 172.16.22.1: bytes=32 time<10ms TTL=127
Reply from 172.16.22.1: bytes=32 time<10ms TTL=127
Reply from 172.16.22.1: bytes=32 time<10ms TTL=127
Reply from 172.16.22.1: bytes=32 time<10ms TTL=127

Ping statistics for 172.16.22.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum =  0ms, Average =  0ms
```

▪ **Audit result**
· Test environment is independent and IP address is assigned properly. · **PASS !**

Item ID - 5	Baseline
▪ **Testing**	
· step 1. – Run sniffer-inbound & sniffer-outbound (capture mode) · step 2. – Run Fragrouter (baseline) · step 3. – Attempt to web connection from client to server · step 4. – Confirm whether connection between client and server was established. If not connected, stop all programs and procedures, then try again from step 1 repeatedly · step 5. – Stop sniffer-inbound & sniffer-outbound · step 6. – Compare session log between sniffer-inbound and sniffer-outbound - Packets were transmitted as stated above through Fragrouter ? - If not as intended, think what is the wrong about testing procedures	
▪ **Expected result**	
 Extranet Intranet eth0 eth1 Dummy Hub Fragrouter Dummy Hub Victim (Server - Web) Attacker (Client) Session flow Sniffer - inbound Sniffer - outbound Inbound Packet Log Outbound Packet Log IP TCP DATA	
▪ **Actual result**	
· step 1. – Run sniffer-inbound & sniffer-outbound (capture mode) *Go to the File > New Capture > Select Adapter > Choose the Network Adapter > Check Promiscuous Mode > Setup User-defined filter “src 172.16.21.3 && dst 172.16.22.3” like shown below and then click the [OK] button*	



User-defined filter:

src 172.16.21.3 && dst 172.16.22.3

Select filter:

- Mac layer
- Network layer
- Transport layer
- Application layer

Current Source

3Com EtherLink PCI - ₩Device₩Packet_{E49B5FA5-4325-4F88-A45I

Select Adapter...

☑ Promiscuous Mode

□ Overwrite current capture document

Adapter's name	Internal name
3Com EtherLink PCI	₩Device₩Packet_{E49B5FA5-4325-4F88-A450-E...
NdisWan Adapter	₩Device₩Packet_NdisWanIp

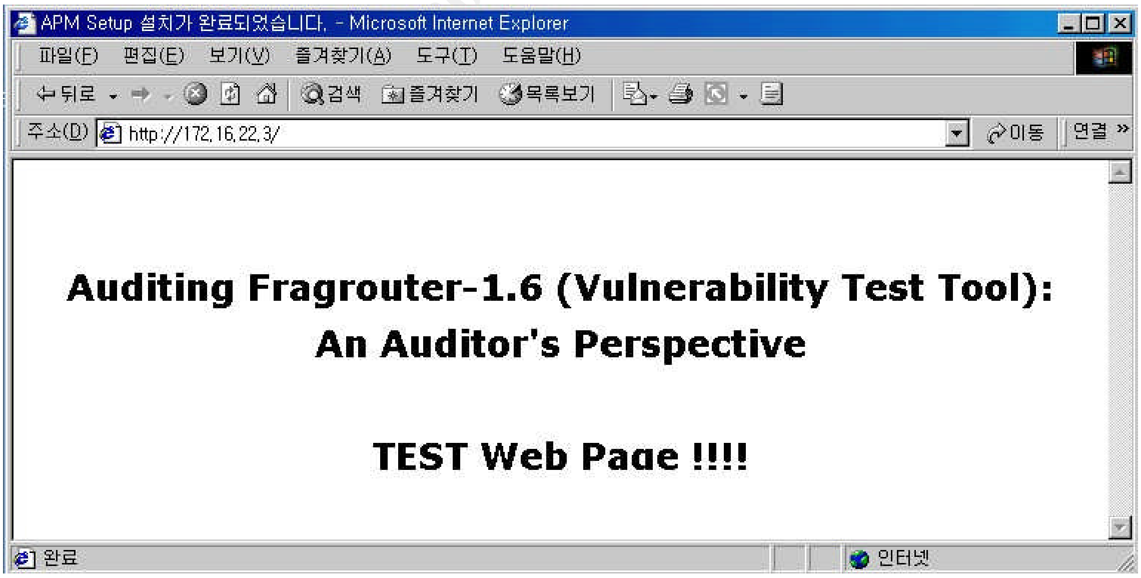
Current Network Adapter

3Com EtherLink PCI

· step 2. – Run Fragrouter as below

./fragrouter –i eth0 –B1

· step 3. – Attempt to web connection from client to server


APM Setup 설치가 완료되었습니다. - Microsoft Internet Explorer
파일(F) 편집(E) 보기(V) 즐겨찾기(A) 도구(T) 도움말(H)
뒤로 검색 즐겨찾기 목록보기
주소(D)
이동
연결
완료
인터넷

http://172.16.22.3/

Auditing Fragrouter-1.6 (Vulnerability Test Tool): An Auditor's Perspective

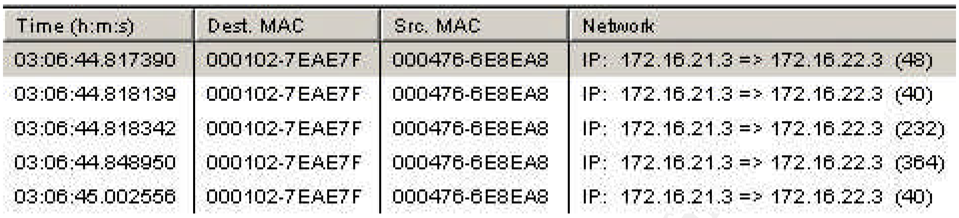
TEST Web Page !!!!

· step 4. – Confirm whether connection between client and server was established. If not connected, stop all programs and procedures, then try again from step 1 repeatedly

· step 5. – Stop sniffer-inbound & sniffer-outbound

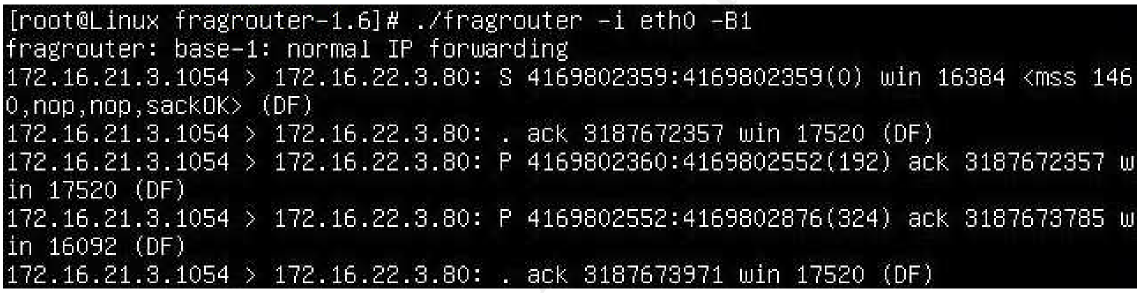
· step 6. – Compare session log between sniffer- inbound and sniffer- outbound

- Sniffer–outbound log



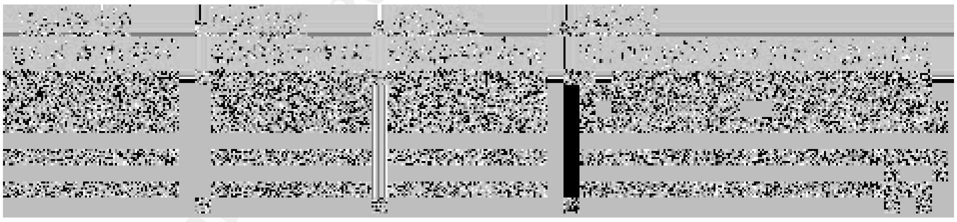
Time (h:m:s)	Dest. MAC	Src. MAC	Network
03:06:44.817390	000102-7EAE7F	000476-6E8EA8	IP: 172.16.21.3 => 172.16.22.3 (48)
03:06:44.818139	000102-7EAE7F	000476-6E8EA8	IP: 172.16.21.3 => 172.16.22.3 (40)
03:06:44.818342	000102-7EAE7F	000476-6E8EA8	IP: 172.16.21.3 => 172.16.22.3 (232)
03:06:44.848950	000102-7EAE7F	000476-6E8EA8	IP: 172.16.21.3 => 172.16.22.3 (364)
03:06:45.002556	000102-7EAE7F	000476-6E8EA8	IP: 172.16.21.3 => 172.16.22.3 (40)

- Fragrouter–log



```
[root@Linux fragrouter-1.6]# ./fragrouter -i eth0 -B1
fragrouter: base-1: normal IP forwarding
172.16.21.3.1054 > 172.16.22.3.80: S 4169802359:4169802359(0) win 16384 <mss 146
0,nop,nop,sackOK> (DF)
172.16.21.3.1054 > 172.16.22.3.80: . ack 3187672357 win 17520 (DF)
172.16.21.3.1054 > 172.16.22.3.80: P 4169802360:4169802552(192) ack 3187672357 w
in 17520 (DF)
172.16.21.3.1054 > 172.16.22.3.80: P 4169802552:4169802876(324) ack 3187673785 w
in 16092 (DF)
172.16.21.3.1054 > 172.16.22.3.80: . ack 3187673971 win 17520 (DF)
```

- Sniffer–outbound log



- Packets were transmitted as stated above through Fragrouter ?
- If not as intended, think what is the wrong about testing procedures.

▪ **Audit result**

· Packets were transmitted as stated above through Fragrouter.

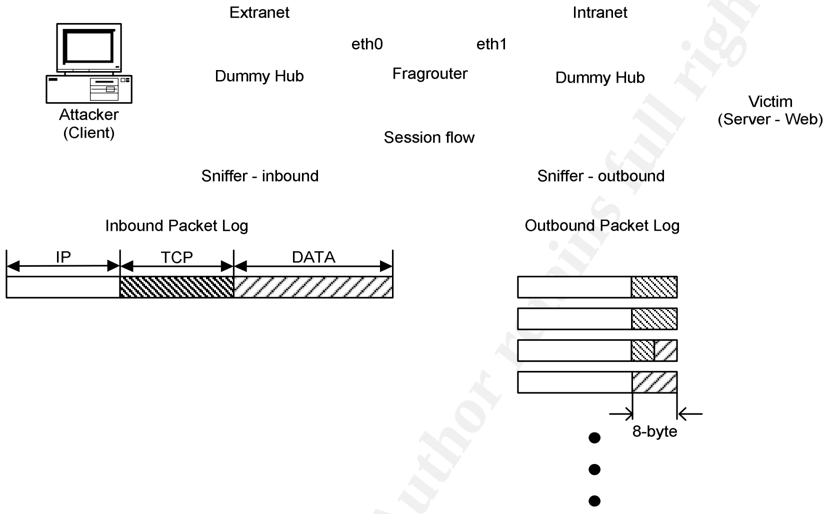
· **PASS !**

Item ID - 6	Frag - 1

▪ Testing

· Same as Item ID - 5

▪ Expected result



▪ Actual result

· step 1. – Run sniffer-inbound & sniffer-outbound (capture mode)

Same as Item - 5

· step 2. – Run Fragrouter as below

```
./fragrouter –i eth0 –F1
```

· step 3. – Attempt to web connection from client to server

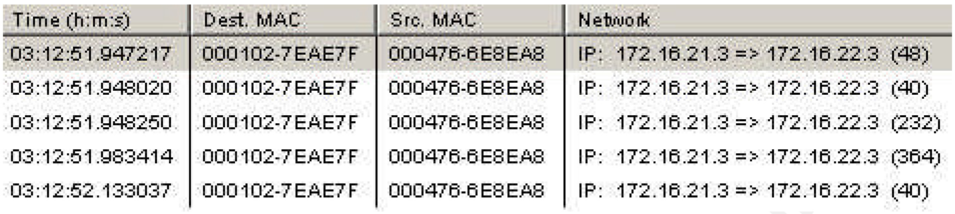
Same as Item - 5

· step 4. – Confirm whether connection between client and server was established. If not connected, stop all programs and procedures, then try again from step 1 repeatedly

· step 5. – Stop sniffer- inbound & sniffer- outbound

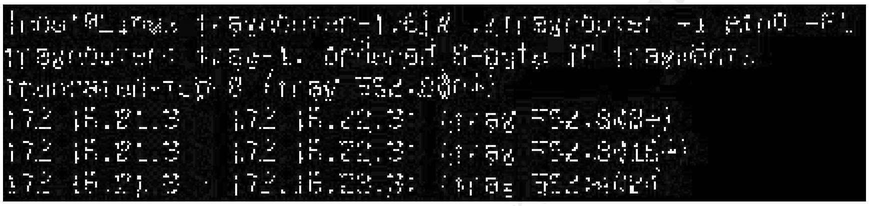
· step 6. – Compare session log between sniffer- inbound and sniffer- outbound

- Sniffer– inbound log



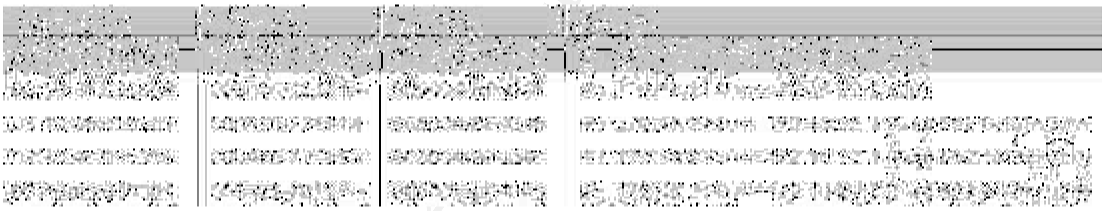
Time (h:m:s)	Dest. MAC	Src. MAC	Network
03:12:51.947217	000102-7EAE7F	000476-6E8EA8	IP: 172.16.21.3 => 172.16.22.3 (48)
03:12:51.948020	000102-7EAE7F	000476-6E8EA8	IP: 172.16.21.3 => 172.16.22.3 (40)
03:12:51.948250	000102-7EAE7F	000476-6E8EA8	IP: 172.16.21.3 => 172.16.22.3 (232)
03:12:51.983414	000102-7EAE7F	000476-6E8EA8	IP: 172.16.21.3 => 172.16.22.3 (364)
03:12:52.133037	000102-7EAE7F	000476-6E8EA8	IP: 172.16.21.3 => 172.16.22.3 (40)

- Fragrouter–log



```
[root@linux fragrouter-1.6]# ./fragrouter -i eth0 -F1
fragrouter: frag-1: ordered 8-byte IP fragments
[illegible] 8 (frag 552:8@0+)
172.16.21.3 > 172.16.22.3: (frag 552:8@8+)
172.16.21.3 > 172.16.22.3: (frag 552:8@16+)
172.16.21.3 > 172.16.22.3: (frag 552:8@24)
```

- Sniffer– outbound log



[illegible]

- Packets were transmitted as stated above through Fragrouter ?
- If not as intended, think what is the wrong about testing procedures.

▪ Audit result

· Packets were transmitted as stated above through Fragrouter.

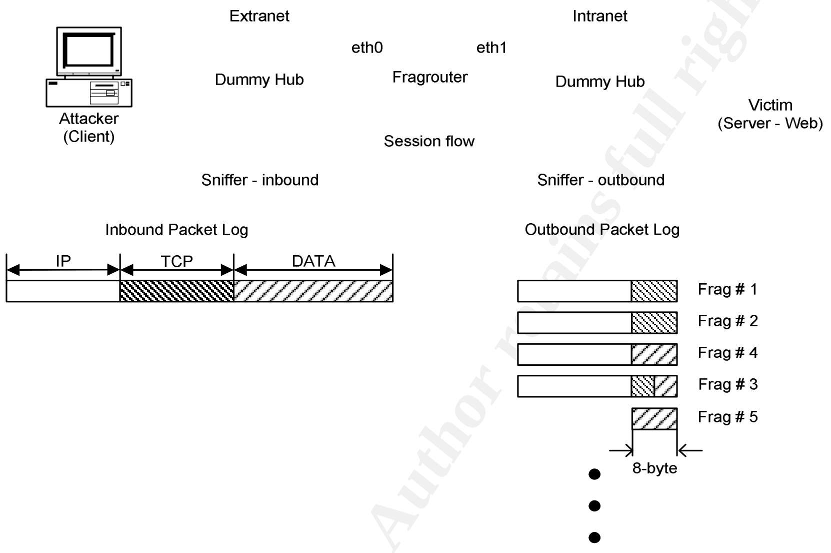
· **PASS !**

Item ID - 8	Frag - 3

▪ Testing

· *Same as Item ID -5*

▪ Expected result



▪ Actual result

· step 1. – Run sniffer-inbound & sniffer-outbound (capture mode)

Same as Item - 5

· step 2. – Run Fragrouter as below

./fragrouter –i eth0 –F3

· step 3. – Attempt to web connection from client to server

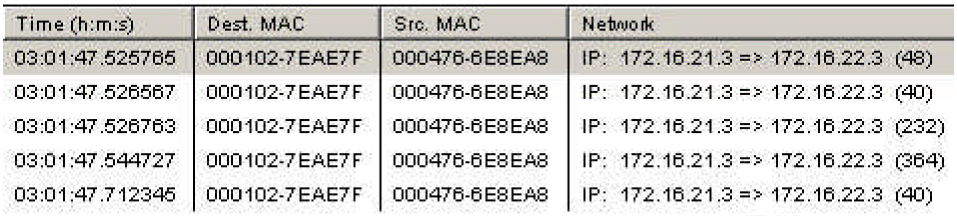
Same as Item - 5

· step 4. – Confirm whether connection between client and server was established. If not connected, stop all programs and procedures, then try again from step 1 repeatedly

· step 5. – Stop sniffer- inbound & sniffer- outbound

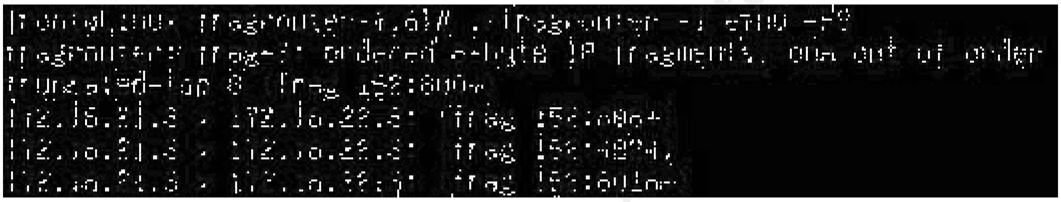
· step 6. – Compare session log between sniffer- inbound and sniffer- outbound

- Sniffer–inbound log



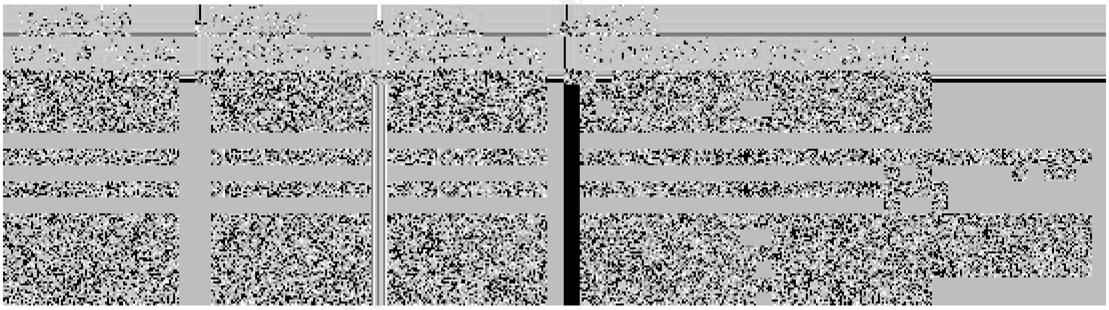
Time (h:m:s)	Dest. MAC	Src. MAC	Network
03:01:47.525765	000102-7EAE7F	000476-6E8EA8	IP: 172.16.21.3 => 172.16.22.3 (48)
03:01:47.526567	000102-7EAE7F	000476-6E8EA8	IP: 172.16.21.3 => 172.16.22.3 (40)
03:01:47.526763	000102-7EAE7F	000476-6E8EA8	IP: 172.16.21.3 => 172.16.22.3 (232)
03:01:47.544727	000102-7EAE7F	000476-6E8EA8	IP: 172.16.21.3 => 172.16.22.3 (364)
03:01:47.712345	000102-7EAE7F	000476-6E8EA8	IP: 172.16.21.3 => 172.16.22.3 (40)

- Fragrouter–log



```
[root@linux fragrouter-1.6]# ./fragrouter -i eth0 -F3
fragrouter: frag-3: ordered 8-byte IP fragments, one out of order
truncated-ip 8 (frag 152:8@0+
172.16.21.3 > 172.16.22.3: (frag 152:8@8+
172.16.21.3 > 172.16.22.3: (frag 152:8@24+
172.16.21.3 > 172.16.22.3: (frag 152:8@16+
```

- Sniffer–outbound log



- Packets were transmitted as stated above through Fragrouter ?
- If not as intended, think what is the wrong about testing procedures.

▪ **Audit result**

· Packets were transmitted as stated above through Fragrouter.

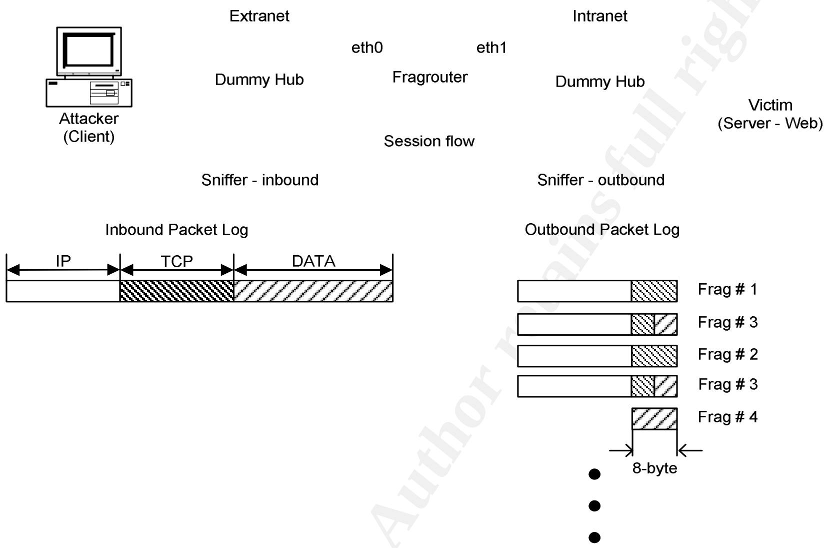
· **PASS !**

Item ID - 10	**Frag - 5**

▪ **Testing**

· ***Same as Item ID -5***

▪ **Expected result**



▪ **Actual result**

· step 1. – Run sniffer-inbound & sniffer-outbound (capture mode)

Same as Item - 5

· step 2. – Run Fragrouter as below

./fragrouter –i eth0 –F5

· step 3. – Attempt to web connection from client to server

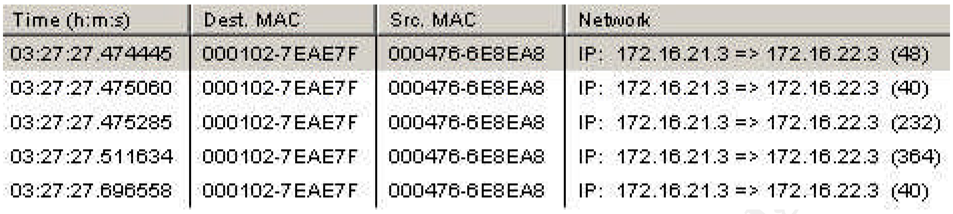
Same as Item - 5

· step 4. – Confirm whether connection between client and server was established. If not connected, stop all programs and procedures, then try again from step 1 repeatedly

· step 5. – Stop sniffer- inbound & sniffer- outbound

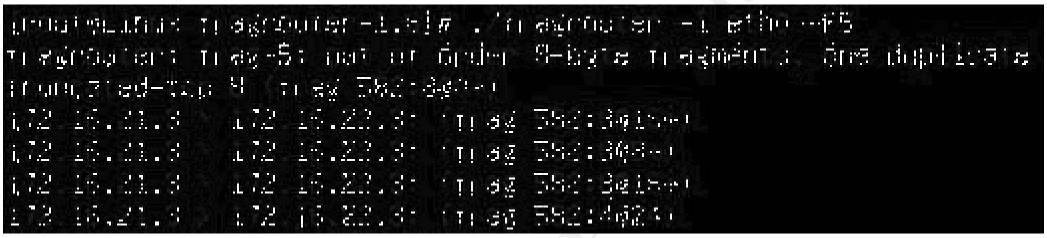
· step 6. – Compare session log between sniffer- inbound and sniffer- outbound

- Sniffer–inbound log



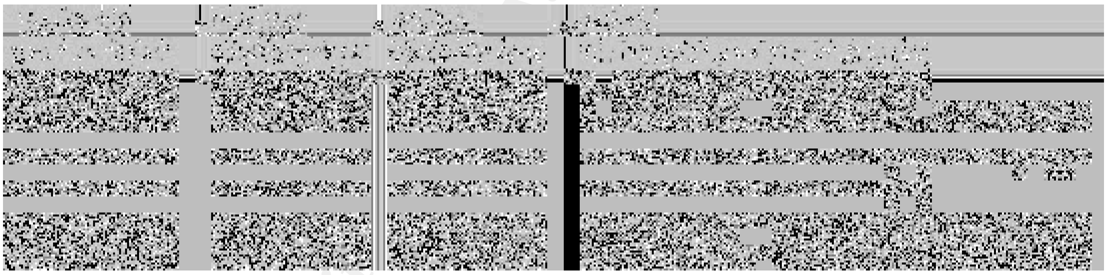
Time (h:m:s)	Dest. MAC	Src. MAC	Network
03:27:27.474445	000102-7EAE7F	000476-6E8EA8	IP: 172.16.21.3 => 172.16.22.3 (48)
03:27:27.475060	000102-7EAE7F	000476-6E8EA8	IP: 172.16.21.3 => 172.16.22.3 (40)
03:27:27.475285	000102-7EAE7F	000476-6E8EA8	IP: 172.16.21.3 => 172.16.22.3 (232)
03:27:27.511634	000102-7EAE7F	000476-6E8EA8	IP: 172.16.21.3 => 172.16.22.3 (364)
03:27:27.696558	000102-7EAE7F	000476-6E8EA8	IP: 172.16.21.3 => 172.16.22.3 (40)

- Fragrouter–log



```
[illegible] fragrouter-1.6]# ./fragrouter -i eth0 -F5
fragrouter: frag-5: out of order 8-byte fragments, one duplicate
truncated-tcp 8 (frag [illegible])
172.16.21.3 > 172.16.22.3: (frag [illegible])
172.16.21.3 > 172.16.22.3: (frag [illegible])
172.16.21.3 > 172.16.22.3: (frag [illegible])
172.16.21.3 > 172.16.22.3: (frag [illegible])
```

- Sniffer–outbound log



- Packets were transmitted as stated above through Fragrouter ?
- If not as intended, think what is the wrong about testing procedures.

▪ **Audit result**

· Packets were transmitted as stated above through Fragrouter.

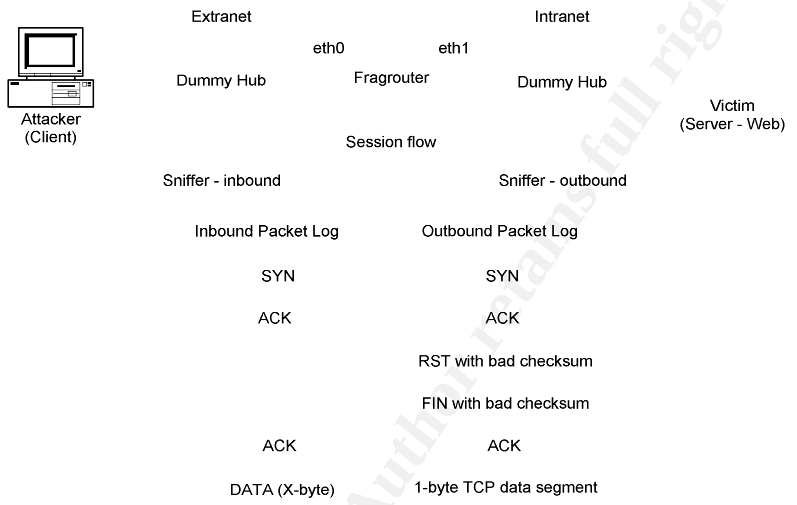
· **PASS !**

Item ID - 13	TCP - 1

▪ **Testing**

· ***Same as Item ID -5***

▪ **Expected result**



▪ **Actual result**

· step 1. – Run sniffer-inbound & sniffer-outbound (capture mode)

Same as Item - 5

· step 2. – Run Fragrouter as below

./fragrouter –i eth0 –T1

· step 3. – Attempt to web connection from client to server

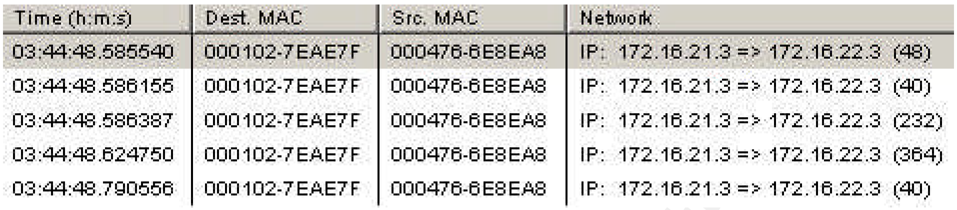
Same as Item - 5

· step 4. – Confirm whether connection between client and server was established. If not connected, stop all programs and procedures, then try again from step 1 repeatedly

▪ step 5. – Stop sniffer- inbound & sniffer- outbound

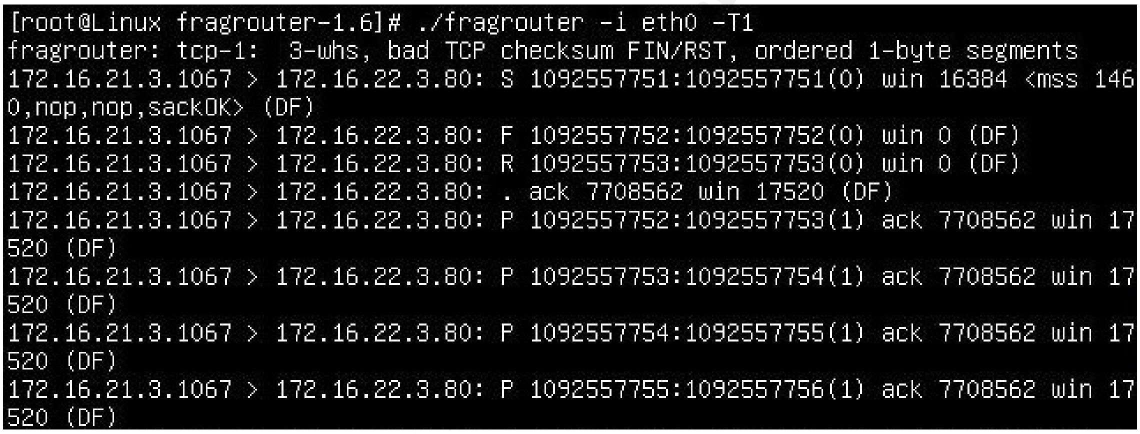
▪ step 6. – Compare session log between sniffer- inbound and sniffer- outbound

- Sniffer–inbound log



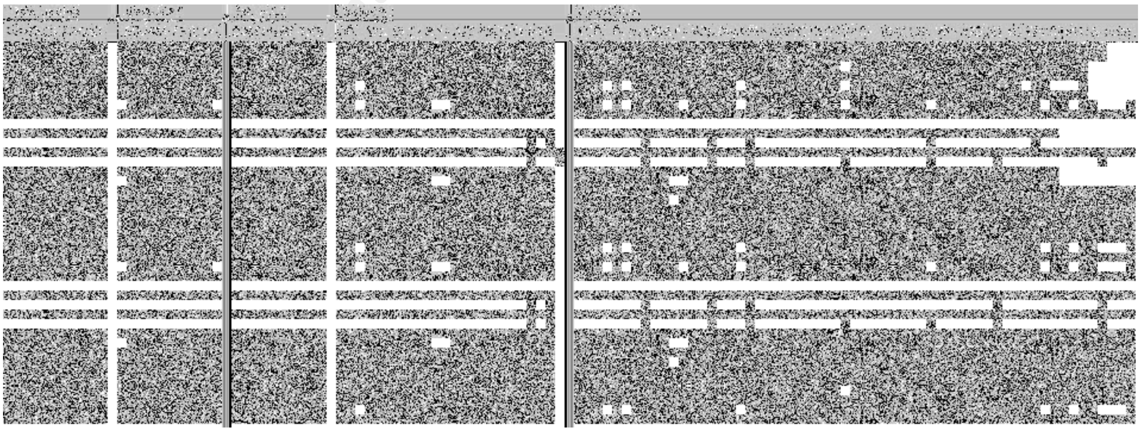
Time (h:m:s)	Dest. MAC	Src. MAC	Network
03:44:48.585540	000102-7EAE7F	000476-6E8EA8	IP: 172.16.21.3 => 172.16.22.3 (48)
03:44:48.586155	000102-7EAE7F	000476-6E8EA8	IP: 172.16.21.3 => 172.16.22.3 (40)
03:44:48.586387	000102-7EAE7F	000476-6E8EA8	IP: 172.16.21.3 => 172.16.22.3 (232)
03:44:48.624750	000102-7EAE7F	000476-6E8EA8	IP: 172.16.21.3 => 172.16.22.3 (364)
03:44:48.790556	000102-7EAE7F	000476-6E8EA8	IP: 172.16.21.3 => 172.16.22.3 (40)

- Fragrouter–log



```
[root@Linux fragrouter-1.6]# ./fragrouter -i eth0 -T1
fragrouter: tcp-1:  3-whs, bad TCP checksum FIN/RST, ordered 1-byte segments
172.16.21.3.1067 > 172.16.22.3.80: S 1092557751:1092557751(0) win 16384 <mss 146
0,nop,nop,sackOK> (DF)
172.16.21.3.1067 > 172.16.22.3.80: F 1092557752:1092557752(0) win 0 (DF)
172.16.21.3.1067 > 172.16.22.3.80: R 1092557753:1092557753(0) win 0 (DF)
172.16.21.3.1067 > 172.16.22.3.80: . ack 7708562 win 17520 (DF)
172.16.21.3.1067 > 172.16.22.3.80: P 1092557752:1092557753(1) ack 7708562 win 17
520 (DF)
172.16.21.3.1067 > 172.16.22.3.80: P 1092557753:1092557754(1) ack 7708562 win 17
520 (DF)
172.16.21.3.1067 > 172.16.22.3.80: P 1092557754:1092557755(1) ack 7708562 win 17
520 (DF)
172.16.21.3.1067 > 172.16.22.3.80: P 1092557755:1092557756(1) ack 7708562 win 17
520 (DF)
```

- Sniffer–outbound log



- Packets were transmitted as stated above through Fragrouter ?
- If not as intended, think what is the wrong about testing procedures.

▪ **Audit result**

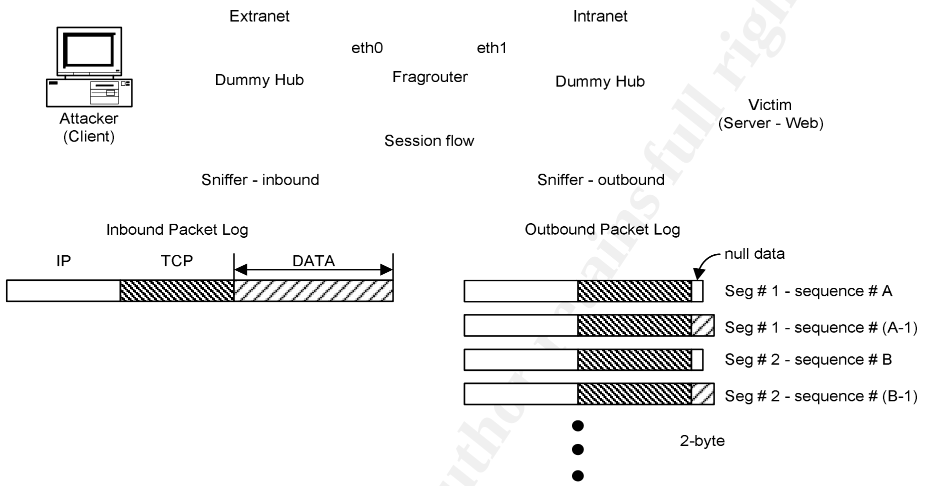
- Packets were transmitted as stated above through Fragrouter.
- **PASS !**

Item ID - 16	TCP - 5

▪ Testing

· *Same as Item ID -5*

▪ Expected result



▪ Actual result

· step 1. – Run sniffer-inbound & sniffer-outbound (capture mode)

Same as Item ID -5

· step 2. – Run Fragrouter as below

```
./fragrouter –i eth0 –T5
```

· step 3. – Attempt to web connection from client to server

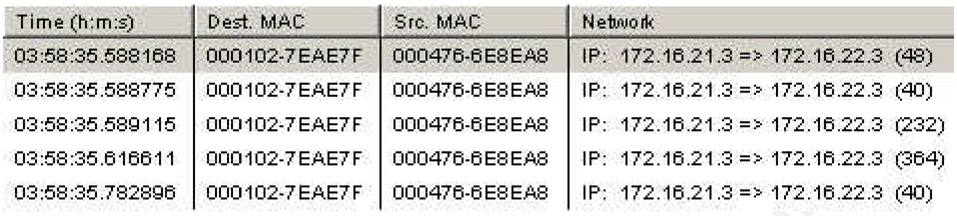
Same as Item ID -5

· step 4. – Confirm whether connection between client and server was established. If not connected, stop all programs and procedures, then try again from step 1 repeatedly

· step 5. – Stop sniffer- inbound & sniffer- outbound

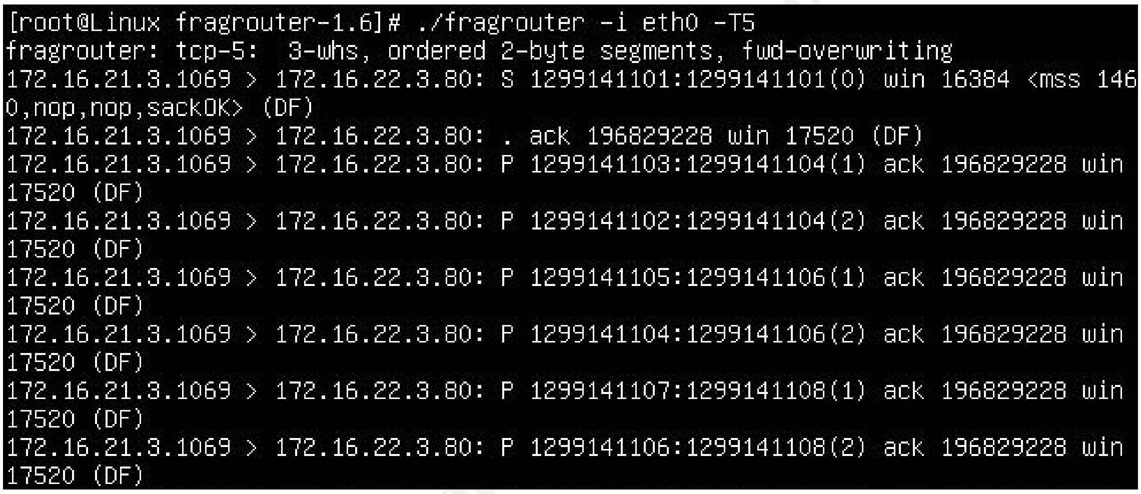
· step 6. – Compare session log between sniffer- inbound and sniffer- outbound

- Sniffer–inbound log



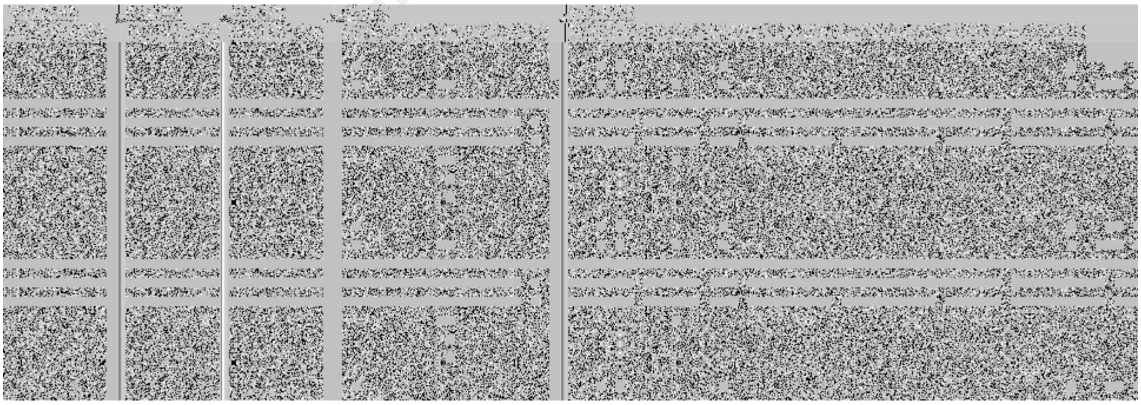
Time (h:m:s)	Dest. MAC	Src. MAC	Network
03:58:35.588168	000102-7EAE7F	000476-6E8EA8	IP: 172.16.21.3 => 172.16.22.3 (48)
03:58:35.588775	000102-7EAE7F	000476-6E8EA8	IP: 172.16.21.3 => 172.16.22.3 (40)
03:58:35.589115	000102-7EAE7F	000476-6E8EA8	IP: 172.16.21.3 => 172.16.22.3 (232)
03:58:35.616611	000102-7EAE7F	000476-6E8EA8	IP: 172.16.21.3 => 172.16.22.3 (364)
03:58:35.782896	000102-7EAE7F	000476-6E8EA8	IP: 172.16.21.3 => 172.16.22.3 (40)

- Fragrouter–log



```
[root@Linux fragrouter-1.6]# ./fragrouter -i eth0 -T5
fragrouter: tcp-5:  3-whs, ordered 2-byte segments, fwd-overwriting
172.16.21.3.1069 > 172.16.22.3.80: S 1299141101:1299141101(0) win 16384 <mss 146
0,nop,nop,sackOK> (DF)
172.16.21.3.1069 > 172.16.22.3.80: . ack 196829228 win 17520 (DF)
172.16.21.3.1069 > 172.16.22.3.80: P 1299141103:1299141104(1) ack 196829228 win
17520 (DF)
172.16.21.3.1069 > 172.16.22.3.80: P 1299141102:1299141104(2) ack 196829228 win
17520 (DF)
172.16.21.3.1069 > 172.16.22.3.80: P 1299141105:1299141106(1) ack 196829228 win
17520 (DF)
172.16.21.3.1069 > 172.16.22.3.80: P 1299141104:1299141106(2) ack 196829228 win
17520 (DF)
172.16.21.3.1069 > 172.16.22.3.80: P 1299141107:1299141108(1) ack 196829228 win
17520 (DF)
172.16.21.3.1069 > 172.16.22.3.80: P 1299141106:1299141108(2) ack 196829228 win
17520 (DF)
```

- Sniffer– outbound log

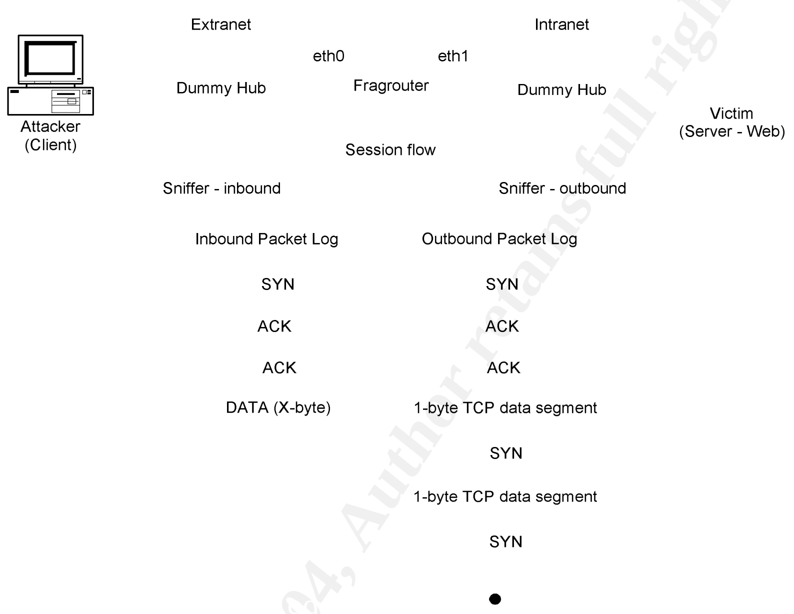


- Packets were transmitted as stated above through Fragrouter ?
- If not as intended, think what is the wrong about testing procedures.

▪ **Audit result**

- Packets were transmitted as stated above through Fragrouter.
- **PASS !**

Item ID - 20	TCBC - 2
▪ **Testing**	
· ***Same as Item ID -5***	
▪ **Expected result**	



▪ **Actual result**

· step 1. – Run sniffer- inbound & sniffer- outbound (capture mode)

Refer Item – 5

· step 2. – Run Fragrouter as below

./fragrouter –i eth0 –C2

· step 3. – Attempt to web connection from client to server

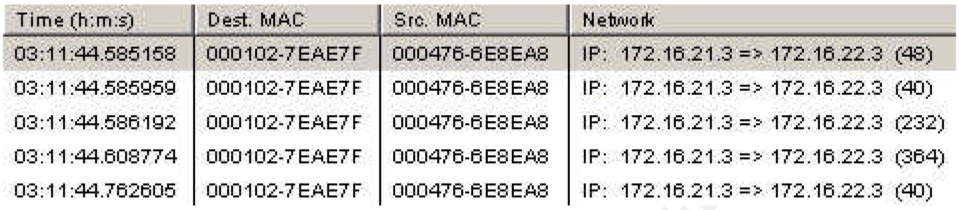
Refer Item ID – 5

· step 4. – Confirm whether connection between client and server was established. If not connected, stop all programs and procedures, then try again from step 1 repeatedly

- step 5. – Stop sniffer-inbound & sniffer-outbound

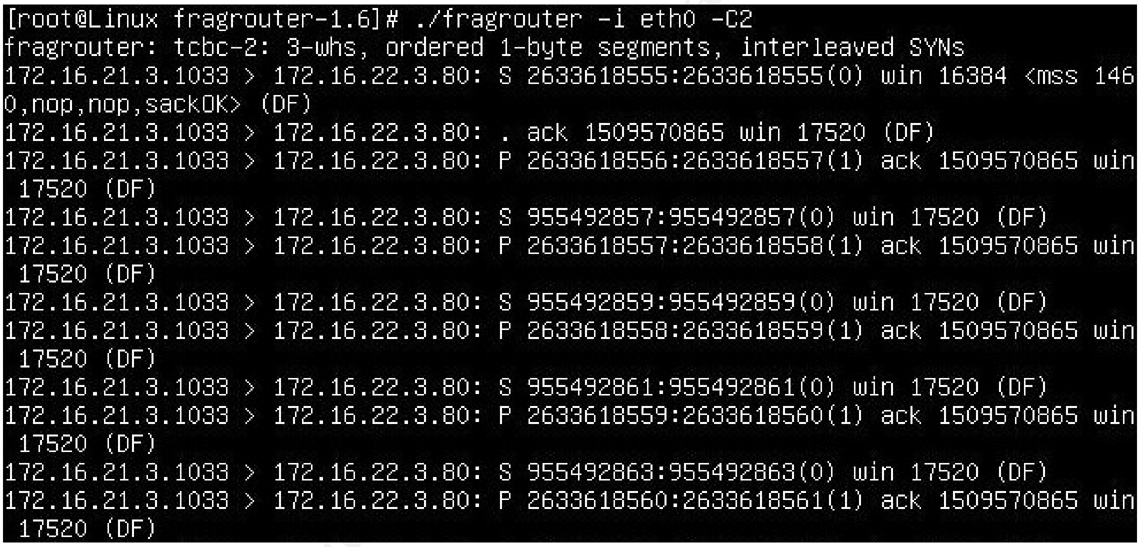
- step 6. – Compare session log between sniffer- inbound and sniffer- outbound

- Sniffer–inbound log



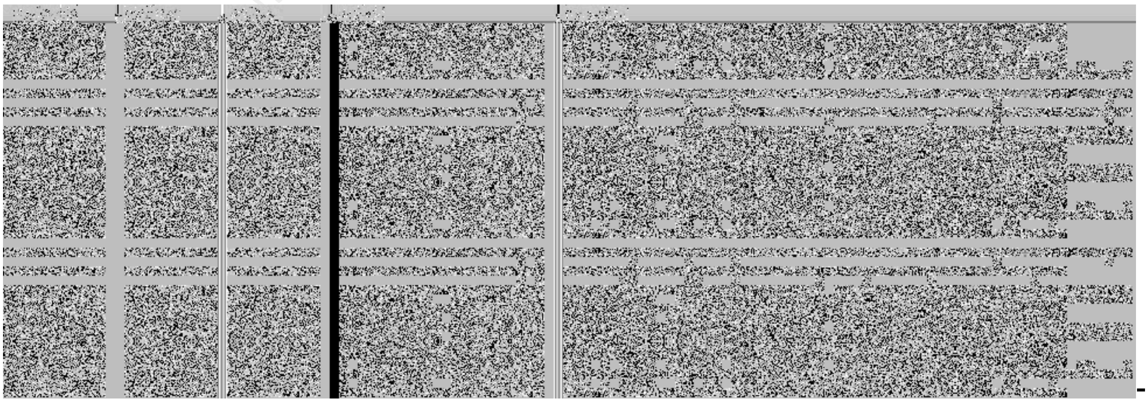
Time (h:m:s)	Dest. MAC	Src. MAC	Network
03:11:44.585158	000102-7EAE7F	000476-6E8EA8	IP: 172.16.21.3 => 172.16.22.3 (48)
03:11:44.585959	000102-7EAE7F	000476-6E8EA8	IP: 172.16.21.3 => 172.16.22.3 (40)
03:11:44.586192	000102-7EAE7F	000476-6E8EA8	IP: 172.16.21.3 => 172.16.22.3 (232)
03:11:44.608774	000102-7EAE7F	000476-6E8EA8	IP: 172.16.21.3 => 172.16.22.3 (364)
03:11:44.762605	000102-7EAE7F	000476-6E8EA8	IP: 172.16.21.3 => 172.16.22.3 (40)

- Fragrouter–log



```
[root@Linux fragrouter-1.6]# ./fragrouter -i eth0 -C2
fragrouter: tcbc-2: 3-whs, ordered 1-byte segments, interleaved SYNs
172.16.21.3.1033 > 172.16.22.3.80: S 2633618555:2633618555(0) win 16384 <mss 146
0,nop,nop,sackOK> (DF)
172.16.21.3.1033 > 172.16.22.3.80: . ack 1509570865 win 17520 (DF)
172.16.21.3.1033 > 172.16.22.3.80: P 2633618556:2633618557(1) ack 1509570865 win
 17520 (DF)
172.16.21.3.1033 > 172.16.22.3.80: S 955492857:955492857(0) win 17520 (DF)
172.16.21.3.1033 > 172.16.22.3.80: P 2633618557:2633618558(1) ack 1509570865 win
 17520 (DF)
172.16.21.3.1033 > 172.16.22.3.80: S 955492859:955492859(0) win 17520 (DF)
172.16.21.3.1033 > 172.16.22.3.80: P 2633618558:2633618559(1) ack 1509570865 win
 17520 (DF)
172.16.21.3.1033 > 172.16.22.3.80: S 955492861:955492861(0) win 17520 (DF)
172.16.21.3.1033 > 172.16.22.3.80: P 2633618559:2633618560(1) ack 1509570865 win
 17520 (DF)
172.16.21.3.1033 > 172.16.22.3.80: S 955492863:955492863(0) win 17520 (DF)
172.16.21.3.1033 > 172.16.22.3.80: P 2633618560:2633618561(1) ack 1509570865 win
 17520 (DF)
```

- Sniffer–outbound log



-

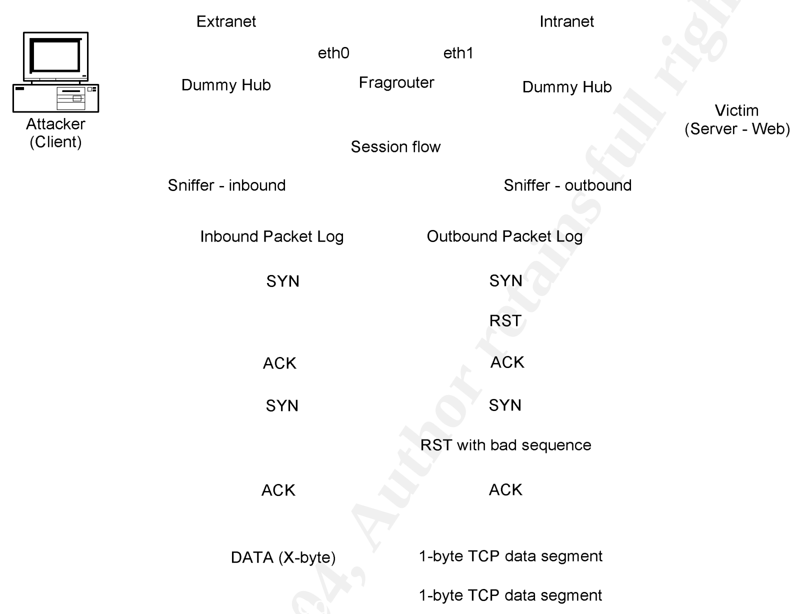
- Packets were transmitted as stated above through Fragrouter ? - If not as intended, think what is the wrong about testing procedures.
▪ **Audit result**
· Packets were transmitted as stated above through Fragrouter. · **PASS !**

Item ID - 22	TCBT - 1

▪ Testing

· Same as Item ID -5

▪ Expected result



▪ Actual result

· step 1. – Run sniffer-inbound & sniffer-outbound (capture mode)

Same as Item ID -5

· step 2. – Run Fragrouter as below

./fragrouter –i eth0 –R1

· step 3. – Attempt to web connection from client to server

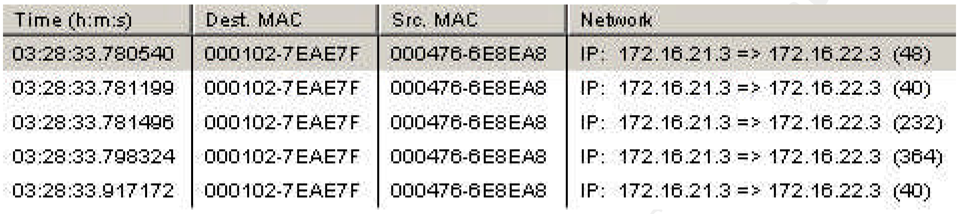
Same as Item ID -5

· step 4. – Confirm whether connection between client and server was established. If not connected, stop all programs and procedures, then try again from step 1 repeatedly

· step 5. – Stop sniffer- inbound & sniffer- outbound

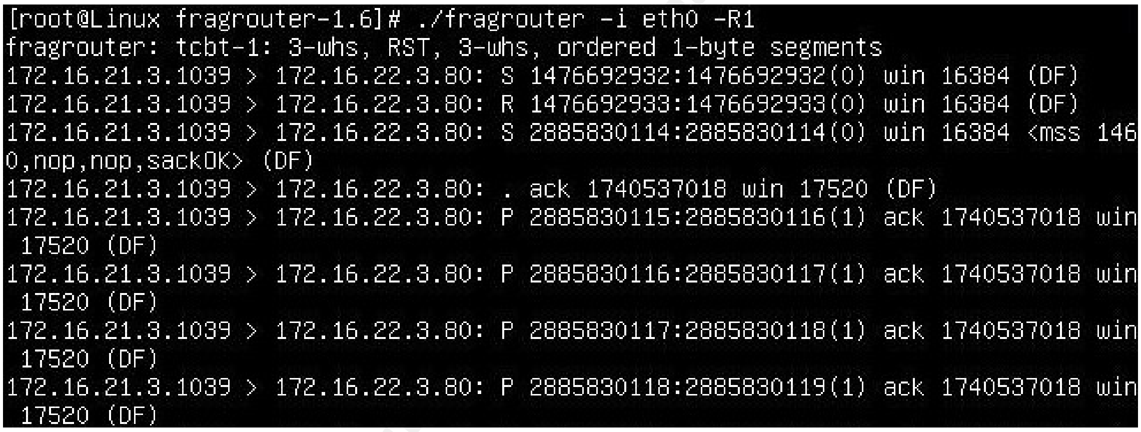
· step 6. – Compare session log between sniffer- inbound and sniffer- outbound

- Sniffer–inbound log



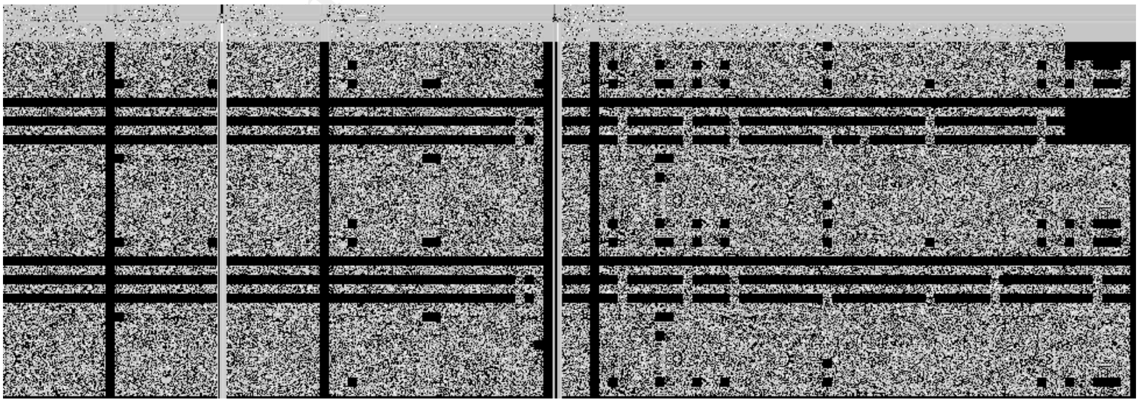
Time (h:m:s)	Dest. MAC	Src. MAC	Network
03:28:33.780540	000102-7EAE7F	000476-6E8EA8	IP: 172.16.21.3 => 172.16.22.3 (48)
03:28:33.781199	000102-7EAE7F	000476-6E8EA8	IP: 172.16.21.3 => 172.16.22.3 (40)
03:28:33.781496	000102-7EAE7F	000476-6E8EA8	IP: 172.16.21.3 => 172.16.22.3 (232)
03:28:33.798324	000102-7EAE7F	000476-6E8EA8	IP: 172.16.21.3 => 172.16.22.3 (364)
03:28:33.917172	000102-7EAE7F	000476-6E8EA8	IP: 172.16.21.3 => 172.16.22.3 (40)

- Fragrouter–log



```
[root@Linux fragrouter-1.6]# ./fragrouter -i eth0 -R1
fragrouter: tcbt-1: 3-whs, RST, 3-whs, ordered 1-byte segments
172.16.21.3.1039 > 172.16.22.3.80: S 1476692932:1476692932(0) win 16384 (DF)
172.16.21.3.1039 > 172.16.22.3.80: R 1476692933:1476692933(0) win 16384 (DF)
172.16.21.3.1039 > 172.16.22.3.80: S 2885830114:2885830114(0) win 16384 <mss 146
0,nop,nop,sackOK> (DF)
172.16.21.3.1039 > 172.16.22.3.80: . ack 1740537018 win 17520 (DF)
172.16.21.3.1039 > 172.16.22.3.80: P 2885830115:2885830116(1) ack 1740537018 win
 17520 (DF)
172.16.21.3.1039 > 172.16.22.3.80: P 2885830116:2885830117(1) ack 1740537018 win
 17520 (DF)
172.16.21.3.1039 > 172.16.22.3.80: P 2885830117:2885830118(1) ack 1740537018 win
 17520 (DF)
172.16.21.3.1039 > 172.16.22.3.80: P 2885830118:2885830119(1) ack 1740537018 win
 17520 (DF)
```

- Sniffer–outbound log



- Packets were transmitted as stated above through Fragrouter ?
- If not as intended, think what is the wrong about testing procedures.

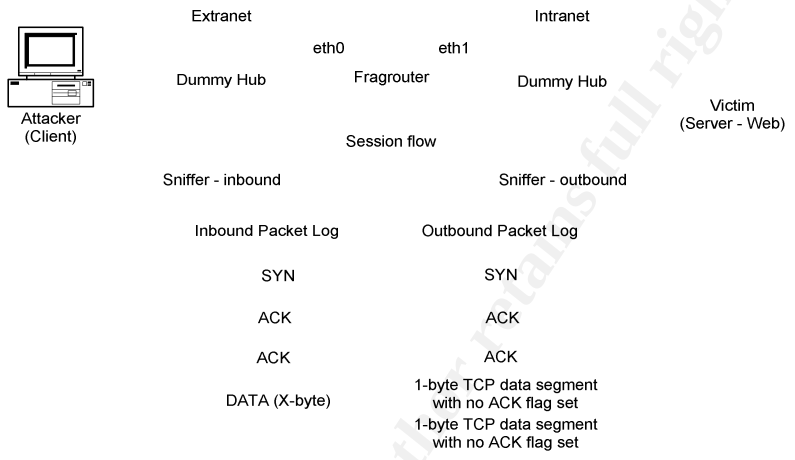
▪ **Audit result**
· Packets were transmitted as stated above through Fragrouter. · **PASS !**

Item ID - 24	INS - 3

▪ Testing

· *Same as Item ID -5*

▪ Expected result



▪ Actual result

· step 1. – Run sniffer-inbound & sniffer-outbound (capture mode)

Same as Item ID -5

· step 2. – Run Fragrouter as below

./fragrouter –i eth0 –I3

· step 3. – Attempt to web connection from client to server

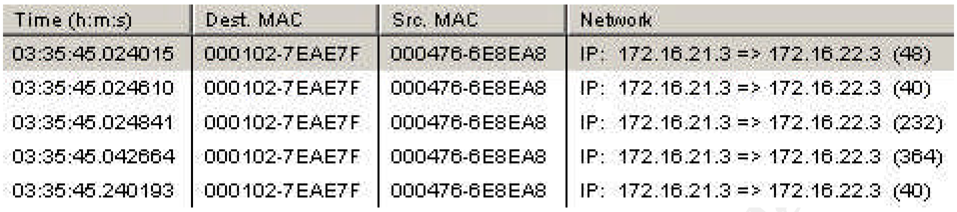
Same as Item ID -5

· step 4. – Confirm whether connection between client and server was established. If not connected, stop all programs and procedures, then try again from step 1 repeatedly

· step 5. – Stop sniffer- inbound & sniffer- outbound

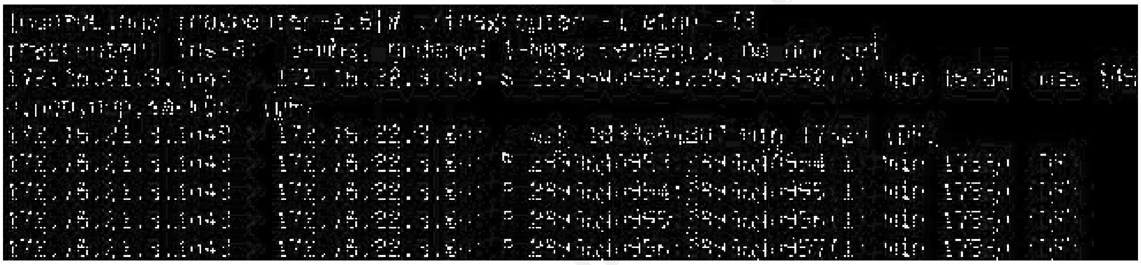
· step 6. – Compare session log between sniffer- inbound and sniffer- outbound

- Sniffer–inbound log



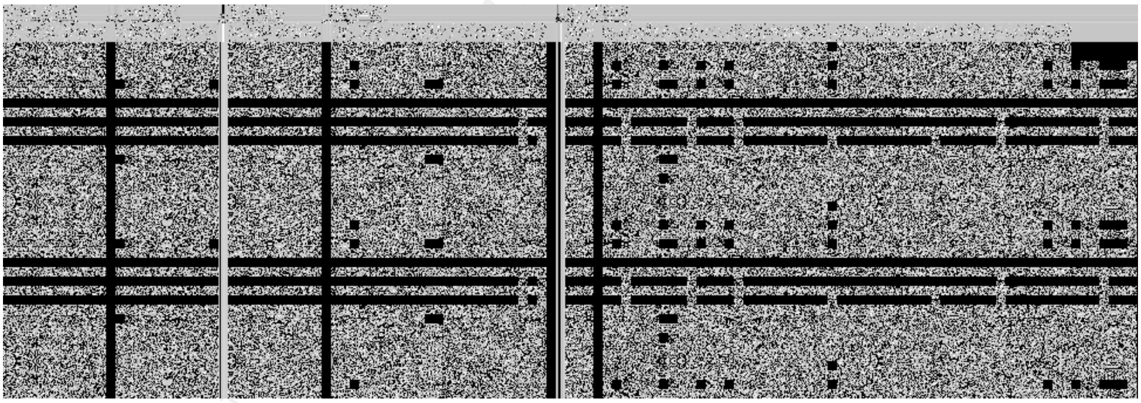
Time (h:m:s)	Dest. MAC	Src. MAC	Network
03:35:45.024015	000102-7EAE7F	000476-6E8EA8	IP: 172.16.21.3 => 172.16.22.3 (48)
03:35:45.024610	000102-7EAE7F	000476-6E8EA8	IP: 172.16.21.3 => 172.16.22.3 (40)
03:35:45.024841	000102-7EAE7F	000476-6E8EA8	IP: 172.16.21.3 => 172.16.22.3 (232)
03:35:45.042664	000102-7EAE7F	000476-6E8EA8	IP: 172.16.21.3 => 172.16.22.3 (364)
03:35:45.240193	000102-7EAE7F	000476-6E8EA8	IP: 172.16.21.3 => 172.16.22.3 (40)

- Fragrouter–log



```
[illegible]
```

- Sniffer–outbound log



- Packets were transmitted as stated above through Fragrouter ?
- If not as intended, think what is the wrong about testing procedures.

▪ **Audit result**

· Packets were transmitted as stated above through Fragrouter.

· **PASS !**

3.2 – Measure Residual Risk

Once risks have been identified, existing controls can be evaluated or new controls designed to reduce the vulnerabilities to an acceptable level of risk. They could be actions, devices, procedures or techniques. The remaining level of risk, once audits and controls have been applied, is called residual risks. Residual risk can be used by management to identify those areas in which more control is required to further reduce risk. A target of an acceptable level of risk can be established by management. Risks in excess of this level should be reduced by the implementation of more stringent controls.

Audit checklist presented in this paper is for Fragrouter that is automatic tool used to evaluate and analyze NIDS's security functionality. Though the majority of the intended control objectives were achieved successfully during the auditing of Fragrouter as stated above, residual risks may be remained. Here are two types of residual risk that must be considered.

One is related to the environmental exposure. There are primarily due to naturally occurring events – fire, natural disasters (earthquake, volcano, hurricane etc), power failure, power spike, air conditioning failure, electrical shock, equipment failure, water damage and so on. Though these can not be eliminated in advance, instead a various detective or corrective can help to mitigate risk. Commonly detective or corrective controls include the following:

- Water detectors
- Hand-held fire extinguishers, Manual fire alarms, Smoke detectors, Fire suppression systems, Fireproof walls
- Electrical surge protectors, Uninterruptible power supply, Power leads from two substations

Another one is related to the operating system. Because operating system can not be implemented perfectly and may have some vulnerability that may affect to the functionality of Fragrouter. More specially, fault code – it can cause buffer overflow, format string, race condition, library error attack – can be used as a threat. In order to minimize these risks, tester must install the latest vulnerability patches and service packs periodically.

Additionally this is not certain residual risk or not, Fragrouter should not be utilized as an attack tool.

3.3 – Is the System Auditable?

This paper will permit comparability among the results of independent security audits. It does so by providing a common set of control objectives for the security functions of vulnerability testing tools (to analyze NIDS's security functionality) and for environment applied to them during a security audit. The security analysis functions of Fragrouter and the environment (physical, procedural, personnel, and network configuration security) items applied to them could be met to control objectives through audit process in the checklist and auditable enough for providing evidence to be used as validation materials. The audit results may help testers to determine whether the vulnerability testing tool is performed properly for their intended application.

We can conclude that this paper is auditable because of following reasons:

□ Environment item :

- The related documents provide the control objectives rationale that describes all the physical, procedural, personnel, and network configuration security measures that are necessary to protect the confidentiality and integrity of the Fragrouter in its test environment.

□ Functionality item :

- The expected test result provides the anticipated output form a successful execution of the test
- The actual test result from the auditor execution of the test is compared with expected test result in order that demonstrate each tested functionality behaved as specified.

Assignment 4 – Audit Report

4.1– Executive Summary

An objective of this paper is for auditing on the FRAGROUTER that is automatic tool to perform vulnerability analysis about NIDS. Also, it focused on identified risks and proposed controls that are directly related result from improper operation and environment of the target system.

As stated in assignment 3, all audit checklist items (even we conducted only 10 items) were passed because actual results are equal to expected results. It means that Fragrouter operated correctly and appropriately as specified or desired demands aspects of environmental and functional requirements. Therefore, Fragrouter can be utilized to the test about NIDS's reliability related fragmentation (segmentation) evasion attacks.

An audit made relative to the proposed checklist items represents the findings of a specific type of investigation of the environmental and functional properties on a vulnerability testing tool. Such audit does not guarantee fitness for use in any particular application environment and additional functional requirements. Additionally, the process of this audit can be applied to audit about other vulnerability testing tool for guaranteeing reliability of NIDS.

In the next chapter, we will think over recommendations that are based on consideration of some security issue including the audit findings.

4.2 - Audit Findings

· Category : Environment

Item No : 1 ~ 4

Auditing Findings :

Item 1 ~ 4 consist of physical, procedural, personnel, and network configuration security, to be used as the basis for audit of environmental properties of the Fragrouter. By guaranteeing such a proposed basis, the results of a Fragrouter's audit will be meaningful to a wider audience. In other words, this requires that the statements resulting from audit are defensible.

Though the performed audit process established a level of confidence that the environmental properties of Fragrouter met the presented control objectives, additional risk related to the system's overload still exist.

Background/Risk :

The system's overload can lead to the probability of function failures in operation. Once a system is in operation, it is possible that Fragrouter can not collect and route network packets because of resource's lack. The followings are root cause related to the packet capture loss.

- An excess of NIC's capacity
- A insufficiency of available memory space
- An excess of CPU's capacity

Audit Recommendations :

Tester (or auditor) should take more care in Fragrouter operation to eliminate risk sated above. Before perform the Fragrouter, the tester shall examine the system's resource through the following methods.

- Check NIC's capacity through manual's specification
- Check packet loss using by ***ifconfig*** command
- Check the usage of memory and CPU using by ***ps, top, sar*** command

Costs :

The commands presented above are internal command supported by operating system and free.

Compensating Controls :

If the tester do not know or use represent commands, he/she would have to consider buying a commercial security resource (for example, SMS(Server Management System) costs approximately $100 over).

· Category : Functionality

Item No : 5 ~ 24

Auditing Findings :

Item 5 ~ 24 consist of baseline, frag, tcp, tcbc, tcbt, and ins test, to be used as the basis for audit of functional properties of the Fragrouter. By guaranteeing such a proposed basis, the results of a Fragrouter's audit will be meaningful to a wider audience. In other words, this requires that the statements resulting from audit are defensible.

Though the performed audit process established a level of confidence that the functional properties of Fragrouter met the presented control objectives, additional risk related to the NIDS's evasion attack still exist.

Background/Risk :

A NIDS's evasion method is a flaw in the security of network and so various. It can not guarantee the NIDS's security reliability to the other evasion attack methods, because Fragrouter only can provide fragmentation (segmentation) evasion attacks. Therefore, an attacker can evade the NIDS through following methods.

- CGI (http protocol) scanning attack
- False positive attack
- Other sophisticated IDS evasion technique attacks

Audit Recommendations :

Tester (or auditor) should take more care to eliminate risk sated above. To increase NID's secure functionality on other evasion attacks except fragmentation attack, the tester can use following tools.

- Test CGI (http protocol) scanning attack using by ***whisker*** tool
- Test False positive attack using by ***IDSwakeup*** tool

Costs :

The tools presented above can be obtain from internet and are freeware.

Compensating Controls :

If the tester want to test more sophisticated IDS evasion technique attacks, he/she would have to consider buying a commercial tool(for example, Hailstorm costs approximately $500 over).

· References

See Sections 1.4 for complete listing of all references