GIAC CERTIFIED INTRUSION ANALYST
PRACTICAL ASSIGNMENT
VERSION 4.0

MARCO BRANDO

DECEMBER 16TH, 2004

TABLE OF CONTENTS

ABSTRACT

Through the following work, we wish to demonstrate the capacity of the students, how to put in practice, in an effective way, the knowledge they have acquired in intrusion detection over computers networks. In order to test this knowledge, an intrusion analysis must be developed in a hypothetical scenario. This analysis must contain the following elements:

1. An executive summary describing in a general way all the process of research done.
2. Establish a detailed analysis of three selected events from a series of log files provided by GIAC.
3. Present a briefing the process used by the analyst to generate the results of the report.

Finally recommendations of appropriate security are made for the analyzed scenario that allows the use of an adequate preventive management of security events that might occur in the future.

PART I. EXECUTIVE SUMMARY

Universities all around the world, as centers of knowledge developers, carry out a multiplicity of investigations in different areas. For that reason, they must have laboratories where they accomplish complex practices. Even though these environments are generally controlled and it would seem dichotomist to comment on it, they should be as open as possible in relation to the freedom of action for the execution of those practices. Join the fact that in recent times with the advances that exist in the communication levels on the network of networks, the Internet, the investigations are accomplished by scientists and laboratories that are located in different countries, in a constant manner through informatics networks, sharing information of the obtained results; finalizing in the publication of the results in their Web Sites in order to be consulted by any party interested in these investigations and results.

Based on the described scenario, the University Campus Intrusion Detectors company (IDUC), specialized in information security and particularly in everything related with the intrusion detection in the field of academics nets, has been selected to achieve a process of audits in a prestigious university. The main reason of the hiring of IUDC is the fact that the university has detected abnormal activities inside their computers networks and presumes that intrusive actions are presenting themselves on their informatics assets.

IDUC, during the process did the analysis of the information given by University Security Department, which consisted of a compile of logs during a determined period of time.

In fact, after studying the information, convincing invasion activities were detected from outer networks logged into the university network that included port scans and worm attacks among others.

IDUC, in the following document, shows a detailed analysis of the findings, the impact it can cause on the networks of the university and specific recommendations to stop these attacks and a proactive way future Mephistophelian activities could be prevented.

In the following sections we show the details of the security audit:

PART II. DETAILED ANALYSIS

1. Selected Scenario

In order to accomplish the analysis three (3) log files were given by the university and were downloaded from the following URL: **http://isc.sans.org/logs/Raw/.** The logs mentioned were:

- 2002.10.14
- 2002.10.15
- 2002.10.16

Based on the files names, it is presumed that they belong to the days 14, 15 and 16 of October 2002, even though the file corresponding with the day 15 have information of the day 14, and the file with the day 16 has information of the day 15. This was established through a analysis of the files and corroborated after seeing the results thrown by Snortsnarf on the files. These log files were generated by Snort operated in a binary mode and were depurated to eliminate from them any information that could compromise the objective of the analysis as it has been shown in the README file located in the URL mentioned above.

2. Relationship between Devices

Based on the information given by the logs, in this section we will put all the pieces that will allows us to complete the puzzle of the logic diagram of the university network. Let's start the task !!!!!

To start the analysis we relied on Ethereal. Initially we used the filter **eth.src != 00:00:0c:04:b2:33 and eth.src != 00:03:e3:d9:26:c0,** to verified the existence of an other MAC address different to the previous ones from were packages could be originated. The result was empty (there were no other addresses) so we concluded that the only addresses origin for all the packages analyzed were the following: **00:00:0c:04:b2:33** and **00:03:e3:d9:26:c0**

Reviewing these MAC addresses during the traffic analysis with Ethereal, our following discovery was that both belong to CISCO so we can graph our first version of connectivity and infer that our sensor is located between these two devices. The image would be as following:

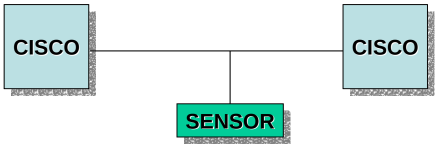
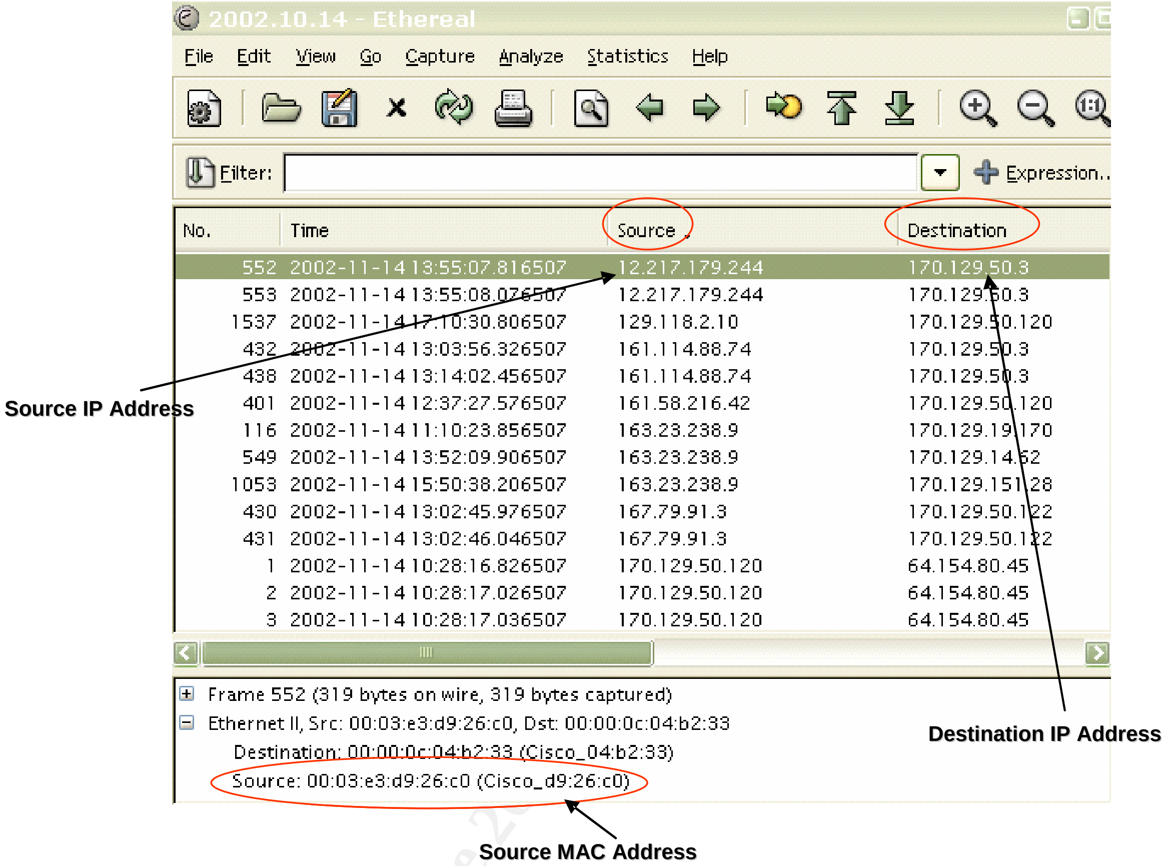


Figure 1

Now we deeply analyze the flow of traffic between both CISCO devices and we can see that the movement of all the packages that have as a MAC source address **00:03:e3:d9:26:c0**, their IP's all correspond to public addresses. That we can see with an example of the Ethereal window shown below.



```
No.   Time                         Source            Destination
 552  2002-11-14 13:55:07.816507   12.217.179.244    170.129.50.3
 553  2002-11-14 13:55:08.076507   12.217.179.244    170.129.50.3
1537  2002-11-14 17:10:30.806507   129.118.2.10      170.129.50.120
 432  2002-11-14 13:03:56.326507   161.114.88.74     170.129.50.3
 438  2002-11-14 13:14:02.456507   161.114.88.74     170.129.50.3
 401  2002-11-14 12:37:27.576507   161.58.216.42     170.129.50.120
 116  2002-11-14 11:10:23.856507   163.23.238.9      170.129.19.170
 549  2002-11-14 13:52:09.906507   163.23.238.9      170.129.14.62
1053  2002-11-14 15:50:38.206507   163.23.238.9      170.129.151.28
 430  2002-11-14 13:02:45.976507   167.79.91.3       170.129.50.122
 431  2002-11-14 13:02:46.046507   167.79.91.3       170.129.50.122
   1  2002-11-14 10:28:16.826507   170.129.50.120    64.154.80.45
   2  2002-11-14 10:28:17.026507   170.129.50.120    64.154.80.45
   3  2002-11-14 10:28:17.036507   170.129.50.120    64.154.80.45
```

```
⊞ Frame 552 (319 bytes on wire, 319 bytes captured)
⊟ Ethernet II, Src: 00:03:e3:d9:26:c0, Dst: 00:00:0c:04:b2:33
    Destination: 00:00:0c:04:b2:33 (Cisco_04:b2:33)
    Source: 00:03:e3:d9:26:c0 (Cisco_d9:26:c0)
```

Figure 2

This pattern repeats itself all through the analysis and revision of the selected log files selected, from which we can establish that the MAC address 00**:03:e3:d9:26:c0** represents the connection to Internet, meanwhile the MAC **00:00:0c:04:b2:33** represents the connection with the internal network. The result would be as following:

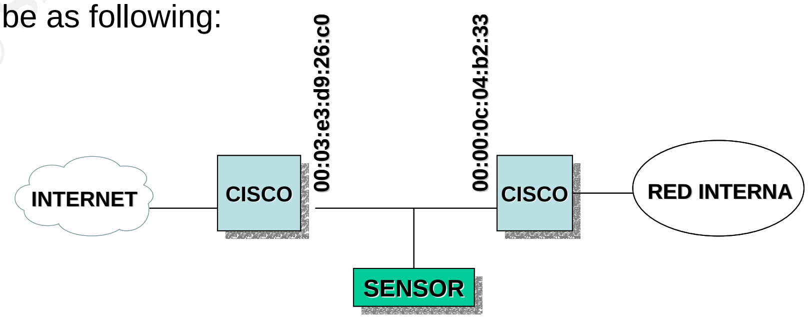


Figure 3

In his investigations, Rob Perdue[1] established that the MAC **00:03:e3:d9:26:c0** belongs to a CISCO PIX firewall, from which that MAC address we presume would be the PIX internal “interface”. We will wait for the detailed analysis of the logs to determine if this premise is totally correct. In relation with the other CISCO device, there is no conclusive information, but based on knowledge of network architecture it could be inferred that it represents a router.

We proceed to study the movement of traffic and the relationship between source and destination addresses to clarify the panorama of the internal network. One simple look at the analyzed log shows that the traffic comes in and out from the network **170.129.0.0/16**. Applying the filter **ip.dst == 170.129.0.0/16** on Ethereal and we can see that the traffic in its majority is HTTP (port 80) so we conclude that this service is offered in the internal network.

Let’s see the outgoing traffic from the internal network using in Ethereal the inverted filter of the last one (**ip.src == 170.129.0.0/16)**. It stands out of the result of applying the filter, that a great percentage of the entries have the origin IP address **170.129.50.120** using the HTTP service (port 80). Analyzing these packages we can observe that a great amount of requests to various external WEB servers exist.

Tying ends, this takes to focus our attention in the address **170.129.50.120** mentioned before. In addition to that, we can observe that the TTL field in the packages whose source address is **170.129.50.120** varies which allows us to deduct that the device with this address is doing NAT labors.

Finally, putting the puzzle pieces all together, the final university network diagram would be the following:

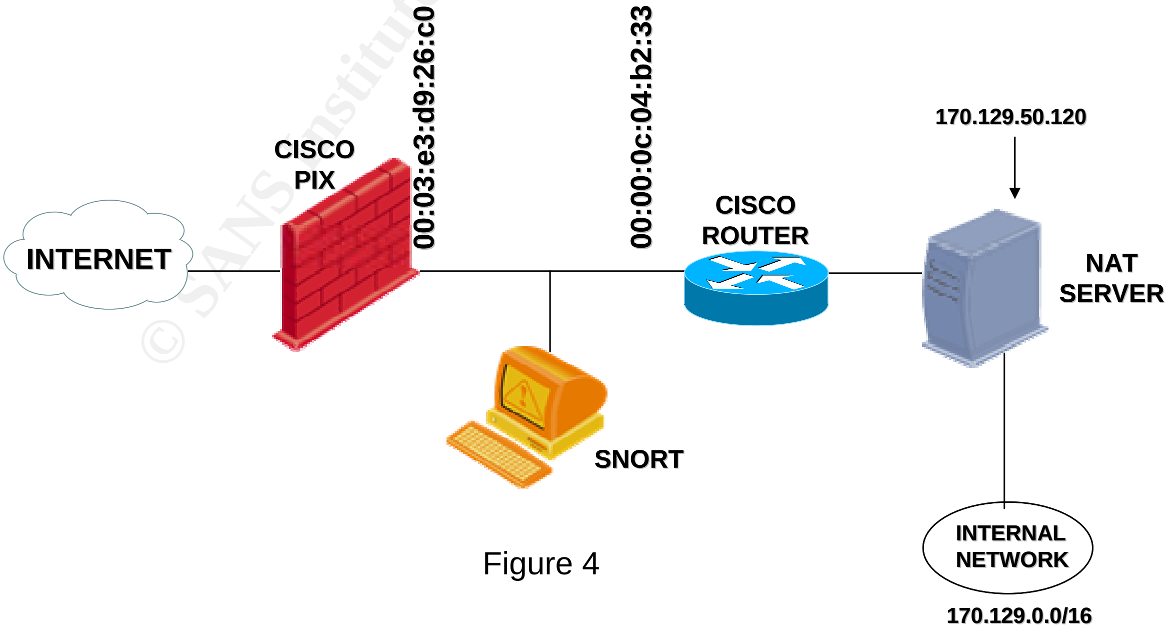


Figure 4

A links graph to illustrate the relationship with the WEB server would be the following (it doesn't show all the relations, it is used as an example):

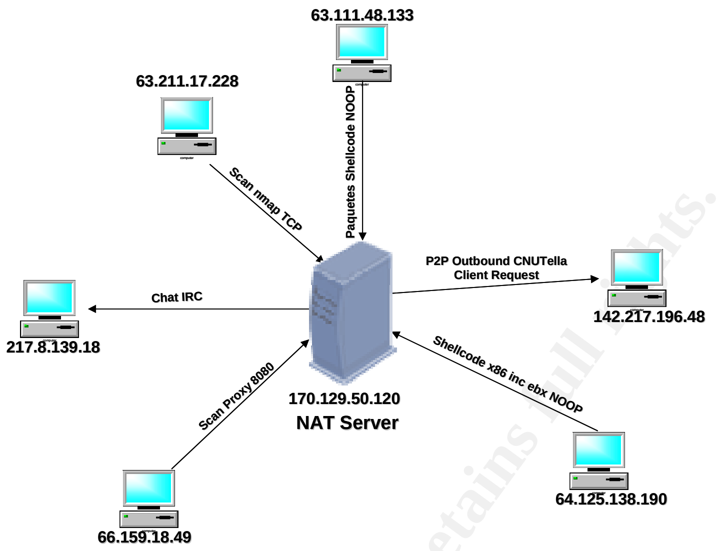


Figure 5

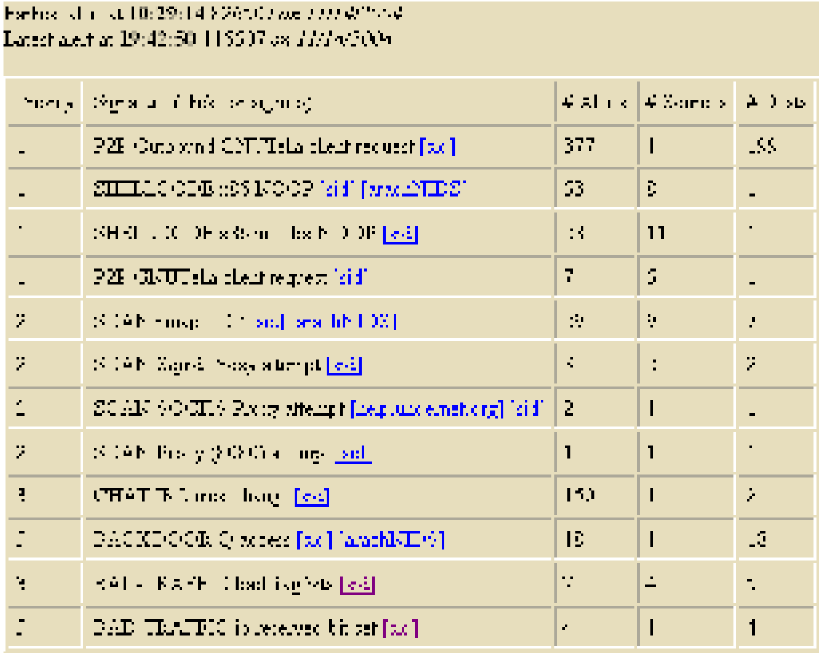
3. Identified Attacks

Through the following charts obtained from Snortsnarf, identified as figure 6, figure 7 and figure 8, all the alerts generated by Snort for the files 2002.10.14, 2002.10.15 and 2002.10.16 respectively can be visualized.

The syntax used with Snortsnarf to generate the results was:

snortsnarf.pl alert.ids –win –rs where:

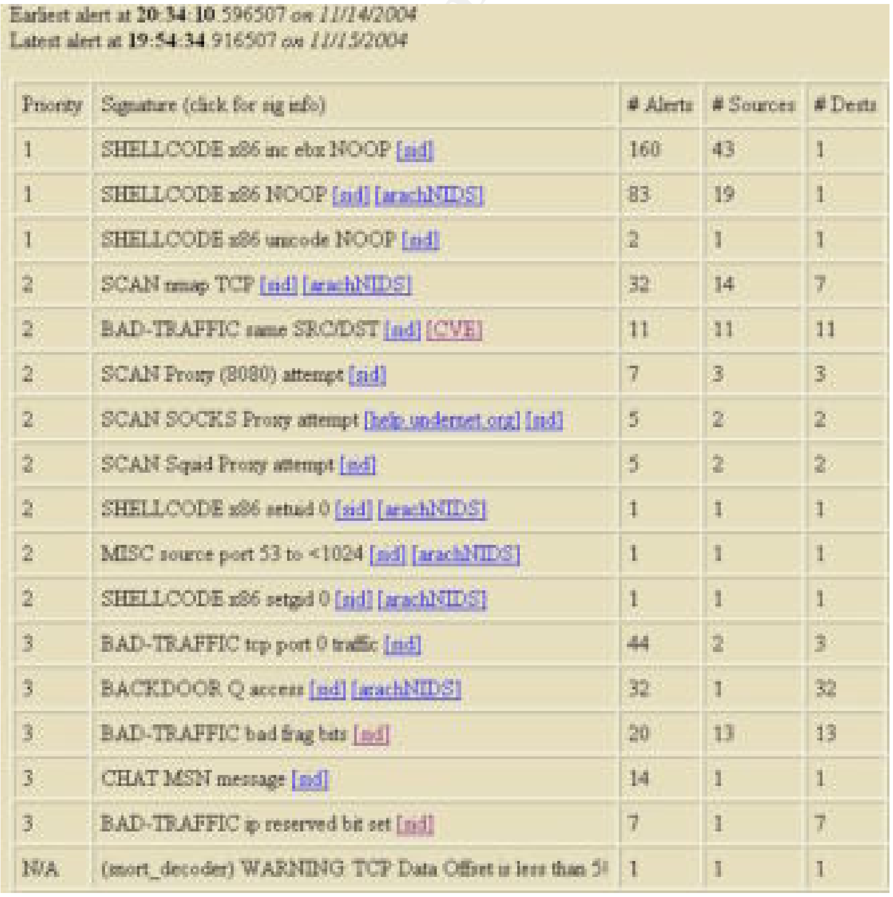
- alert.ids is the file of alerts generated by Snort. It was obtained through the following Snort command –r <source file> -c snort.conf –l <destination> where:
    - –r option that indicates that the information will be read and processed from a tcpdump file.
    - <source file> corresponds to 2002.10.14, 2002.10.15 o 2002.10.16 earlier mentioned
    - -c indicates that a file of rules will be used (snort.conf)
    - -l records the results in a directory
    - <destination> corresponds to a direction path where the file alert.ids will be as a result of the application of the rules of the origin files.
- -win indicates that it is executed in windows mode
- -rs put the most important signatures first



Earliest alert at 10:29:14.[illegible] on [illegible]
Latest alert at 19:42:50.115507 on [illegible]

Priority	Signature (click for sig info)	# Alerts	# Sources	# Dests
1	P2P Outbound GNUTella client request [sid]	377	1	155
1	SHELLCODE x86 NOOP [sid] [arachNIDS]	53	8	1
1	SHELLCODE x86 inc ebx NOOP [sid]	13	11	1
1	P2P GNUTella client request [sid]	7	5	1
2	SCAN nmap TCP [sid] [arachNIDS]	[illegible]	9	[illegible]
2	SCAN Squid Proxy attempt [sid]	[illegible]	[illegible]	2
2	SCAN SOCKS Proxy attempt [help.undernet.org] [sid]	2	1	1
2	SCAN Proxy (8080) attempt [sid]	1	1	1
3	CHAT IRC nick change [sid]	150	1	2
3	BACKDOOR Q access [sid] [arachNIDS]	18	1	18
3	BAD-TRAFFIC bad frag bits [sid]	[illegible]	[illegible]	[illegible]
3	BAD-TRAFFIC ip reserved bit set [sid]	[illegible]	1	1

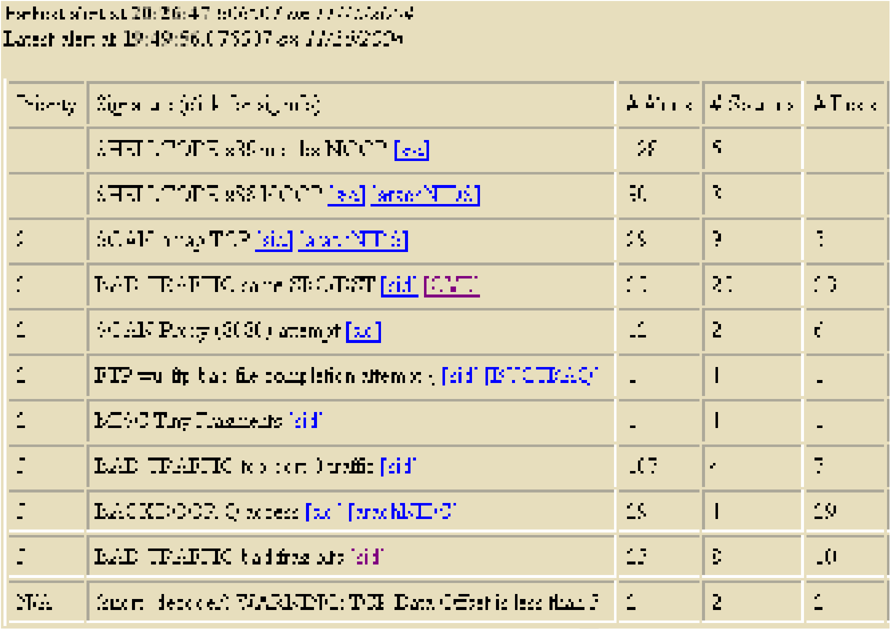
Figure 6



Earliest alert at 20:34:10.596507 on 11/14/2004
Latest alert at 19:54:34.916507 on 11/15/2004

Priority	Signature (click for sig info)	# Alerts	# Sources	# Dests
1	SHELLCODE x86 inc ebx NOOP [sid]	160	43	1
1	SHELLCODE x86 NOOP [sid] [arachNIDS]	83	19	1
1	SHELLCODE x86 unicode NOOP [sid]	2	1	1
2	SCAN nmap TCP [sid] [arachNIDS]	32	14	7
2	BAD-TRAFFIC same SRC/DST [sid] [CVE]	11	11	11
2	SCAN Proxy (8080) attempt [sid]	7	3	3
2	SCAN SOCKS Proxy attempt [help.undernet.org] [sid]	5	2	2
2	SCAN Squid Proxy attempt [sid]	5	2	2
2	SHELLCODE x86 setuid 0 [sid] [arachNIDS]	1	1	1
2	MISC source port 53 to <1024 [sid] [arachNIDS]	1	1	1
2	SHELLCODE x86 setgid 0 [sid] [arachNIDS]	1	1	1
3	BAD-TRAFFIC tcp port 0 traffic [sid]	44	2	3
3	BACKDOOR Q access [sid] [arachNIDS]	32	1	32
3	BAD-TRAFFIC bad frag bits [sid]	20	13	13
3	CHAT MSN message [sid]	14	1	1
3	BAD-TRAFFIC ip reserved bit set [sid]	7	1	7
N/A	(snort_decoder) WARNING: TCP Data Offset is less than 5!	1	1	1

Figure 7



Earliest alert at 20:26:47 [illegible]
Latest alert at 19:49:56.0[illegible]

Priority	Signature (click for sig info)	# Alerts	# Sources	# Dests
	SHELLCODE x86 inc ebx NOOP [sid]	[illegible]	5	
	SHELLCODE x86 NOOP [sid] [arachNIDS]	[illegible]	3	
[illegible]	SCAN nmap TCP [sid] [arachNIDS]	[illegible]	9	[illegible]
[illegible]	BAD-TRAFFIC same SRC/DST [sid] [CVE]	[illegible]	2[illegible]	[illegible]
[illegible]	SCAN Proxy (8080) attempt [sid]	[illegible]	2	[illegible]
[illegible]	FTP wu-ftp bad file completion attempt [sid] [illegible]	[illegible]	1	[illegible]
[illegible]	MISC Tiny Fragments [sid]	[illegible]	1	[illegible]
[illegible]	BAD-TRAFFIC loopback traffic [sid]	[illegible]	[illegible]	[illegible]
[illegible]	BACKDOOR Q access [sid] [arachNIDS]	[illegible]	1	[illegible]
[illegible]	BAD-TRAFFIC bad frag bits [sid]	[illegible]	8	[illegible]
N/A	Snort decoder: WARNING: TCP Data Offset is less than 5	[illegible]	2	[illegible]

Figure 8

3.1 DETECT 1: CODE RED

3.1.1 Description of detect

This attack was located on the file 2002.10.14. In this file, we found some frames with the flag bits "Don't fragment (DF)" and "more fragments (MF)" set in a simultaneous way.

The Windump command used to generate the exit was:

Windump –r 2002.10.14 –nvettttX "ip[6] & 32 ¡=0"

Where:
-r indicates read from a file (in this case 2002.10.14)
-n do not convert the addresses into host names (makes the execution faster)
- v print of determined fields (see Windump help files for more details)
-e print the header of the data link layer (to show the MAC addresses)
-tttt allows to show the event date and time
-X it does the hexadecimal print and it also does it in ASCIl.
"ip[6] &32 ¡=0" looks only for fragmented packets.

The alert generated by Snort was the following:

```
[**] [1:1322:5] BAD-TRAFFIC bad frag bits [**]
[Classification: Misc activity] [Priority: 3]
```

```
11/14-19:42:50.116507 0:3:E3:D9:26:C0 -> 0:0:C:4:B2:33 type:0x800 len:0x5CA
213.107.87.140 -> 170.129.249.190 TCP TTL:111 TOS:0x0 ID:9581 IpLen:20
DgmLen:1468 DF MF  Frag Offset: 0x0000   Frag Size: 0x05A8
```

The letters in bold allow to appreciate in an immediate way the incongruence represented by the indication of the bits DF and MF set in a simultaneous way.

The previous alert corresponds to the following rule:

```
alert ip $EXTERNAL_NET any -> $HOME_NET any (msg:”BAD-TRAFFIC bad
frag bits”; fragbits:MD; classtype:misc-activity; sid:1322; rev:7;)
```

So far we have only mentioned the fragmentation or no fragmentation of the packets, but where is the Code Red? When a more detailed analysis of the packet was done, a great amount of repeated N character (NNNNNNNN) which Is an indication of a Code Red presence, which was corroborated when the packet was studied in detail and compare it with some existing worm references, which are detailed in the following sections. Next, we can see a portion of the packet that generates the malicious traffic:

```
11/14/2002 19:42:50.116507 0:3:e3:d9:26:c0 0:0:c:4:b2:33 0800 1482:
213.107.87.140.3656 > 170.129.249.190.80: P [bad tcp cksum 3fe3!]
227581833:227583261(1428) ack 24679804 win 17520 (frag 9581:1448@0+) (ttl 111,
len 1468)
0x0000   4500 05bc 256d 6000 6f06 ef96 d56b 578c        E...%m`.o....Kw.
0x0010   aa81 f9be 0e48 0050 0d90 9f89 0178 957c        .....H.P.....x.|
0x0020   5018 4470 a71e 0000 4745 5420 2f64 6566        P.Dp....GET./def
0x0030   6175 6c74 2e69 6461 3f4e 4e4e 4e4e 4e4e        ault.ida?NNNNNNN
0x0040   4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e        NNNNNNNNNNNNNNNN
0x0050   4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e        NNNNNNNNNNNNNNNN
0x0060   4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e        NNNNNNNNNNNNNNNN
0x0070   4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e        NNNNNNNNNNNNNNNN
0x0080   4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e        NNNNNNNNNNNNNNNN
0x0090   4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e        NNNNNNNNNNNNNNNN
0x00a0   4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e        NNNNNNNNNNNNNNNN
0x00b0   4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e        NNNNNNNNNNNNNNNN
0x00c0   4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e        NNNNNNNNNNNNNNNN
0x00d0   4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e        NNNNNNNNNNNNNNNN
0x00e0   4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e        NNNNNNNNNNNNNNNN
0x00f0   4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e        NNNNNNNNNNNNNNNN
0x0100   4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e        NNNNNNNNNNNNNNNN
0x0110   4e4e 4e4e 4e4e 4e4e 4e25 7539 3039 3025        NNNNNNNNN%u9090%
0x0120   7536 3835 3825 7563 6264 3325 7537 3830        u6858%ucbd3%u780
0x0130   3125 7539 3039 3025 7536 3835 3825 7563        1%u9090%u6858%uc
0x0140   6264 3325 7537 3830 3125 7539 3039 3025        bd3%u7801%u9090%
0x0150   7536 3835 3825 7563 6264 3325 7537 3830        u6858%ucbd3%u780
```

```
0x0160   3125 7539 3039 3025 7539 3039 3025 7538       1%u9090%u9090%u8
0x0170   3139 3025 7530 3063 3325 7530 3030 3325       190%u00c3%u0003%
0x0180   7538 6230 3025 7535 3331 6225 7535 3366       u8b00%u531b%u53f
0x0190   6625 7530 3037 3825 7530 3030 3025 7530       f%u0078%u0000%u0
0x01a0   303d 6120 2048 5454 502f 312e 300d 0a43      0=a..HTTP/1.0..C
0x01b0   6f6e 7465 6e74 2d74 7970 653ª 2074 6578      ontent-type:.tex
0x01c0   742f 786d 6c0a 484f 5354 3ª77 7777 2e77     t/xml.HOST:www.w
0x01d0   6f72 6d2e 636f 6d0a 2041 6363 6570 743ª      orm.com..Accept:
```

The alteration of the bits DF and MF probably was done to mask the worm and try to pass undetected through possible existing security controls. Review Security Incidents: Initial analysis of the .ida “Code Red” Worm[2].

For the particular case of Code Red, Snort could have generated alerts through some of the following rules:

WEB-IIS ISAPI .ida attempt”; flow:to_server,established; uricontent:”.ida?”; nocase; reference:arachnids,552; classtype:web-application-attack; reference:bugtraq,1065; reference:cve,CAN-2000-0071; sid:1243; rev:8;)

alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:”WEB-IIS ISAPI .ida access”; uricontent:”.ida”; nocase; flow:to_server,established; reference:arachnids,552; classtype:web-application-activity; reference:cve,CAN-2000-0071; reference:bugtraq,1065; sid:1242; rev:6;)

This Code Red attack is identified under CVE-2001-0500 (http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2001-0500)

3.1.2 Reason this detect was selected

This event presented itself 7 times in the monitored period of time; it has been one of the most lethal known against the networks (more than 250,000 infected systems in less than 24 hours in its highest level of spreading). Additionally you can see the packet crafting and the intention to compromise critical resources like the Web servers.

3.1.3 Detect was Generated by

The equipment used to make the detection had Snort (2.0.4 Build 97), Windump 3.6.2, winpcap 2.3, ethereal 0.10.5a. The alerts reading was done with TextPad 4.7.3 and the graphs was obtained with Snortsnarf 0211111.1 (Perl 5.6 is required).

The command used to generate the alert was:

```
 snort –r 2002.10.14 –c snort.conf –l 2002.10.14 –e –d where:

          -c to read the configuration file
          -l destination results folder
          -e shows the information related with the data link layer
          -d dumps the application layer information
```

The rule and the generated alerts were mentioned in the previous section.

3.1.4 Probability the Source Address was Spoofed

Although it is certain that there is always a possibility of spoofed an address, in this particular case we lean to say that it was not spoofed, given the fact that the Code Red in order to operate needs to establish valid communication sessions (3 way handshake)[4]. In addition to this, all the consulted sources about this worm do not reveal that in any given moment it has been detected acting under the mode of spoofed addresses.

3.1.5 Attack Mechanism

The Code Red worm is a malicious worm that attacks the Microsoft IIS Web Servers to which associated security patches have not been applied to protect them from such worm. The worm generates a situation of buffer overflow over vulnerability on the file idq.dll of the Microsoft Index Server which allows the code execution in the user SYSTEM context of the local environment of the compromised server. Later, it uses the involved server to attack other vulnerable Web servers.

The worm tries to establish a connection with the port 80 of the destination host randomly chosen (it must be a Web server). Once the connection with the port 80 is accomplished, the server that is attacked sends a HTTP GET manually altered to the victim, trying to explode a buffer overflow in the Indexing Service. If the attack is effective, the affected Web server will display a message HELLO! Welcome to http://www.worm.com! Hacked By Chinese!.

Another type of activity developed by the worm is associated with the dates displayed on the victim server. When the dates are between the 20 and 28 of each month, the worm tries a DDoS against the government site www1.whitehouse.gov[5].

3.1.6 Correlations

Much documentation exists about Code Red. Nevertheless, those references we considered more relevant are the following (the order in which they appear does not indicate any type of relevance):
Internet Security Systems (ISS): http://xforce.iss.net/xforce/alerts/id/advise89
Microsoft : http://www.microsoft.com/technet/security/bulletin/MS01-033.asp[6]
Trend Micro :
http://www.trendmicro.com/vinfo/virusencyclo/default5.asp?VName=CODERED.A&VSect=T[5]
CVE : http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2001-0500[7]
Eeye: http://www.eeye.com/html/Research/Advisories/AD20010618.html[8]
Cert : http://www.cert.org/advisories/CA-2001-19.html[9]
SANS: Josh Berry GCIA v 4.0 Practical Assignment[4]

3.1.7 Evidence of Active Targeting

Based on the Code Red behavior in which target hosts are randomly selected and verifying the file 2002.10.14, where only in one opportunity a packet from IP 213.107.87.140 was sent to IP 170.129.249.190 which only appears one time too, we can infer that the targets selection was just as the Code Red works, done randomly, reason for which the target is represented by an active host

3.1.8 Severity

Severity = (criticality + lethality) – (system countermeasures + network countermeasures)

Severity = (3+5) – (3+3). Severity = 2

Criticality (3) there is no sufficient information to determine the importance of the target host to the university. Let's remember that the source and destination addresses of the packet that generated the alert, shows up 7 times in the log. It is more important to mention that Code Red is a threat of very high level.

Lethality (5) Code Red generates severe direct damage to the servers that become victims in its operation, beside the fact that it spreads very quickly and in a collateral way to the environments where they operate generating problems like networks congestion.

System Countermeasures (3) enough information does not exist that demonstrate in a categorical manner the existence of evidence of countermeasures in the target Host against Code Red.

Network Countermeasures (3) the information packets in which the altered fragmentation bits were detected, indicates the existence of some sort of countermeasures (sensors), more nevertheless it was allowed, that such malicious packets, continued their journey through the network which put in evidence preventive/reactive protection configuration problems in the presence of these types of threats.

3.2 DETECT 2: TCP TRAFFIC TO THE PORT 0 WITH ID 0

3.2.1 Description of Detect

This new intrusion attempt was detected in the files 2002.10.15 and 2002.10.16. In these files we can observe a number of alerts generated by this kind of situations (see figures 7 and 8), 151 alerts in total. The attack was generated by the addresses 211.47.255.20 through 211.47.255.24.

The Windump command used to obtain only the packets that fulfilled with this condition, was the following:

```
Windump –r 2002.10.15 –nvettttX “dst port 0”
```

Where :
“dst port 0”: only look for the packets in which the destination port is 0.
The rest of the options can be seen in section 3.1.1

The generated alert by Snort was the following:

```
[**] [1:524:6] BAD-TRAFFIC tcp port 0 traffic [**]
[Classification: Misc activity] [Priority: 3]
11/15-16:10:02.006507 0:3:E3:D9:26:C0 -> 0:0:C:4:B2:33 type:0x800 len:0x42
211.47.255.23:47620 -> 170.129.23.96:0 TCP TTL:46 TOS:0x0 ID:0 IpLen:20
DgmLen:52 DF ******S* Seq: 0x88C07AEB  Ack: 0x0  Win: 0x16D0  TcpLen: 32
TCP Options (6) => MSS: 1460 NOP NOP SackOK NOP WS: 0
```

In bold letters you can observe, in an immediate way, the reference of traffic sent to a destination port that is reserved, in this case port 0 and the packet identification with the number 0. We will center our analysis on port 0. The previous alert corresponds to the following rule:

```
alert tcp $EXTERNAL_NET any <> $HOME_NET 0 (msg:"BAD-TRAFFIC tcp port 0 traffic"; classtype:misc-activity; sid:524; rev:6;)
```

With what was shown above and under the knowledge that the port 0 by definition is reserved and is redirected by the operating systems to a ephemeral port, we can infer that the packets that present this anomaly have been altered manually (packets crafting)

3.2.2 Reason this detect was selected

The event as mentioned before was registered in 151 opportunities in a period of approximately 48 hours. Key Points for selecting this attack are the followings:

On the day 15, the attack attempt showed up three periods of time during the course of monitoring.

```
11/15/2002 12:34:50.446507 211.47.255.24.41104 > 170.129.195.40.0: tcp
11/15/2002 12:34:53.296507 211.47.255.24.41104 > 170.129.195.40.0: tcp
11/15/2002 12:34:59.466507 211.47.255.24.41104 > 170.129.195.40.0: tcp
11/15/2002 12:35:11.276507 211.47.255.24.41104 > 170.129.195.40.0: tcp
11/15/2002 12:35:22.326507 211.47.255.24.41358 > 170.129.195.40.0: tcp
11/15/2002 12:35:25.406507 211.47.255.24.41358 > 170.129.195.40.0: tcp
11/15/2002 12:35:31.326507 211.47.255.24.41358 > 170.129.195.40.0: tcp
11/15/2002 12:35:54.326507 211.47.255.24.41611 > 170.129.195.40.0: tcp
11/15/2002 12:35:57.316507 211.47.255.24.41611 > 170.129.195.40.0: tcp
11/15/2002 12:36:03.286507 211.47.255.24.41611 > 170.129.195.40.0: tcp
11/15/2002 12:36:15.296507 211.47.255.24.41611 > 170.129.195.40.0: tcp
11/15/2002 12:36:26.406507 211.47.255.24.41866 > 170.129.195.40.0: tcp
11/15/2002 12:36:29.296507 211.47.255.24.41866 > 170.129.195.40.0: tcp
11/15/2002 12:36:35.286507 211.47.255.24.41866 > 170.129.195.40.0: tcp
11/15/2002 12:36:47.306507 211.47.255.24.41866 > 170.129.195.40.0: tcp

11/15/2002 16:56:36.296507 211.47.255.24.42742 > 170.129.21.249.0: tcp
11/15/2002 16:56:39.336507 211.47.255.24.42742 > 170.129.21.249.0: tcp
11/15/2002 16:56:45.296507 211.47.255.24.42742 > 170.129.21.249.0: tcp
11/15/2002 16:56:57.656507 211.47.255.24.42742 > 170.129.21.249.0: tcp
11/15/2002 16:57:08.306507 211.47.255.24.42950 > 170.129.21.249.0: tcp
11/15/2002 16:57:11.266507 211.47.255.24.42950 > 170.129.21.249.0: tcp
11/15/2002 16:57:17.306507 211.47.255.24.42950 > 170.129.21.249.0: tcp
11/15/2002 16:57:29.376507 211.47.255.24.42950 > 170.129.21.249.0: tcp
11/15/2002 16:57:40.276507 211.47.255.24.43155 > 170.129.21.249.0: tcp
11/15/2002 16:57:49.426507 211.47.255.24.43155 > 170.129.21.249.0: tcp
11/15/2002 16:58:01.296507 211.47.255.24.43155 > 170.129.21.249.0: tcp
11/15/2002 16:58:12.296507 211.47.255.24.43338 > 170.129.21.249.0: tcp
11/15/2002 16:58:15.286507 211.47.255.24.43338 > 170.129.21.249.0: tcp
11/15/2002 16:58:21.306507 211.47.255.24.43338 > 170.129.21.249.0: tcp
11/15/2002 16:58:33.276507 211.47.255.24.43338 > 170.129.21.249.0: tcp

11/15/2002 20:08:17.016507 211.47.255.23.46919 > 170.129.23.96.0: tcp
11/15/2002 20:08:19.996507 211.47.255.23.46919 > 170.129.23.96.0: tcp
11/15/2002 20:08:25.996507 211.47.255.23.46919 > 170.129.23.96.0: tcp
11/15/2002 20:08:37.996507 211.47.255.23.46919 > 170.129.23.96.0: tcp
11/15/2002 20:08:49.016507 211.47.255.23.47154 > 170.129.23.96.0: tcp
11/15/2002 20:08:52.016507 211.47.255.23.47154 > 170.129.23.96.0: tcp
11/15/2002 20:09:10.016507 211.47.255.23.47154 > 170.129.23.96.0: tcp
```

```
11/15/2002 20:09:21.006507 211.47.255.23.47382 > 170.129.23.96.0: tcp
11/15/2002 20:09:23.996507 211.47.255.23.47382 > 170.129.23.96.0: tcp
11/15/2002 20:09:29.986507 211.47.255.23.47382 > 170.129.23.96.0: tcp
11/15/2002 20:09:42.006507 211.47.255.23.47382 > 170.129.23.96.0: tcp
11/15/2002 20:09:56.016507 211.47.255.23.47620 > 170.129.23.96.0: tcp
11/15/2002 20:10:02.006507 211.47.255.23.47620 > 170.129.23.96.0: tcp
11/15/2002 20:10:14.016507 211.47.255.23.47620 > 170.129.23.96.0: tcp
```

Later, on day 15 as in day 16:

- Whenever the attack showed up, it lasted around two minutes approximately
- In each period all packets was directed to the same destination IP address
- In each period of attack, the same source port was used three (3) or four (4) times
- The ID of all the packets was 0 which is invalid

All these aspects keep a relationship between them and in which many similarities can be seen, take us to infer in an evident way that the packets were altered and the objective seems to be that thesource is trying some type of recognition activity through the responses generated by the target IP.

Another reason to think the packsets were altered was obtained through the consulting the Joe Bowling's[10] GCIA Practical Assignment, in which is inferred the packets crafting, having as an origin an Internet publication related to this issue[11.]

3.2.3 Detect was Generated by

The equipment used to carry out the detection is the same as section 3.1.3
The Snort command used to generate the alert was:

```
snort –r 2002.10.14 –c snort.conf –l 2002.10.15 –e –d
```

The meaning of the options can be observed in section 3.1.3.
The rules and generated alerts were already mentioned in the previous section.

3.2.4 Probability the source address was spoofed

We particularly think that the source address was not spoofed, due to the fact that the sender seems to be waiting some type of answer by the sent frames. In addition, it is known that the port 0 is used to make fingerprinting on the operating systems.

3.2.5 Attack Mechanism

This attack can be catalogued as a way of Operating System identification (fingerprinting). It is evident that the packets have been manually altered choosing as a destination port 0 and as a sequence number also 0 without this last one increased it value with the passing of packets and time. Additionally, in each burst of sending this type of packets, the destination address is the same and the source port changes every 3 or 4 packets in each attack attempt. This is why we think that the attacker is looking for a way to obtain information that will allow him/her to gain access to the target systems.

According to the RFC 1700[12], port 0 is reserved for special purposes. This in conjunction with the fact that when this port is used, the operating system redirects it to ephemeral ports, allows us to infer that through the Internet this type of traffic should not travel. It is desirable to know over what operating system the attacker is trying to recollect information but Windows as well as UNIX have the same way to responding to these types of "stimulus" on port 0.

The fingerprinting on port 0 consists of four (4) TCP protocol tests (there is a total of seven; the other tests are for UDP protocol). In our case we put each of them that are relevant for the investigation. To find out information about the totality of the tests consult references.[13]

1. Sending TCP packets from port 0 to port 0
2. Sending TCP packets from port X to port 0
3. Sending TCP packets from port 0 to an open port
4. Sending TCP packets from port 0 to a closed port

The standard answers for cases 1, 2 and 4 should be a RST packet. For case number 3 the answer should be a SYK ACK, if the port is open and the port 0 is valid.

3.2.6 Correlations

- Port 0 OS fingerprinting by Ste Jones Networkpenetration.com http://securityfocus.com/archive/129/330750[13]
- SANS GIAC GCIA (intrusion Detection In-Depth) material given in the course (this material is only given to the students actives in the course) http://giactc.giac.org/cgi-bin/momgate IDs: ID_23_1203.pdf, ID_25_1203.pdf, ID_42_1203[14]
- Joe Bowling's GCIA v 3.3 practical[10]
- Eric Evans's GCIA v 3.4 practical[15]

3.2.7 Evidence of Active Targeting

We think that the host is active because the only information packets sent to the target addresses mentioned previously are exactly those where the destination port is 0, with which the attacker is trying to obtain information that will allow identify the operating system that is being executed in the host.

3.2.8 Severity

Severity = (criticality + lethality) – (system countermeasures + network countermeasures)

Severity = (2+3) – (4+3). Severity = -2

Criticality (2) there is not enough information to determine the relevance of the target host to the university. The target addresses only appear in the logs for these types of attacks. There is no evidence of different activities over them.

Lethality (3) as it has been mentioned, this type of attack tries to obtain information about target host operating system. On its own beside providing this type of information, the attack has no mayor achievement; nevertheless with the information obtained, it can be the starting point of more lethal attacks.

System Countermeasures (4) all through the analysis, it can be perceived that the attacker is trying to establish a connection by sending packets to port 0 with SYN flag set, but don't have any answer from target host and puts in evidence some kind of measures taken in the different attacked systems.

Network Countermeasures (3) the packets of information that were altered were detected and the corresponding alerts were generated, which shows the existence of intrusion detection elements, nevertheless the malicious packets were allowed to continue with their journey within the network which puts in evidence problems in the protection configuration against these types of threats or the non existence of perimeter firewalls.

Given the fact that the severity of the event is negative, we can conclude that it does not represent a real threat.

3.3 DETECTION 3: BACKDOOR Q

3.3.1 Description of detect

This anomaly was found in the files of 2002.10.14, 2002.10.15 y 2002.10.16 which contains the information captured between the days November 14 to 16 of

2002. When reviewing the alerts generated by Snort, we detected an important number of notifications generated on this threat. The total number of alerts was seventy nine (79).

We can see that the source address during this attack was always 255.255.255.255, the source port is 31337, and the ID of the packets is 0, with the packets directed to the internal network addresses 170.129.0.0/16 in a random way.

The windump command used to acomplished the logs reading and obtains the packets that comply with this condition was:

```
Windump –r 2002.10.16 –nvvettttX  “ip src 255.255.255.255”
```

The alert generated by Snort was the following:

```
**] [1:184:3] BACKDOOR Q access [**]
[Classification: Misc activity] [Priority: 3]
11/14-10:29:14.826507 0:3:E3:D9:26:C0 -> 0:0:C:4:B2:33 type:0x800 len:0x3C
255.255.255.255:31337 -> 170.129.172.186:515 TCP TTL:15 TOS:0x0 ID:0
IpLen:20 DgmLen:43 ***A*R** Seq: 0x0  Ack: 0x0  Win: 0x0  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS203]
```

In bold letters we can observe what was commented about the IP address and the source port. We only show an alert as a reference.

The previous alert corresponds to the following rule:

```
alert tcp 255.255.255.0/24 any -> $HOME_NET any (msg:"BACKDOOR Q
access"; flow:stateless; dsize:>1; flags:A+; reference:arachnids,203;
classtype:misc-activity; sid:184; rev:7;)
```

3.3.2 Reason this detect was selected

The reason to selection of this event is based on the fact that the source address of all packets is the broadcast address and in additionally, the port 31337 is a port that is known to be used by hackers to perpetrate noxious actions. In conjunction with the previous statement, the identification given by Snort about the alert, specifies that this Trojan allows the stealth information and the possibility to take control of the target host[16].

3.3.3 Detect was Generated by

The detection was triggered by the rule describe in the previous section and such rule resides in the configuration file backdoor.rules

An example of the alarm generators packets is the following:

```
11/14/2002 22:23:24.646507 0:3:e3:d9:26:c0 0:0:c:4:b2:33 0800 60:
255.255.255.255.31337 > 170.129.41.171.515: R [tcp sum ok] 0:3(3) ack 0 wi
n 0 [RST cko] (ttl 15, id 0, len 43)
0x0000   4500 002b 0000 0000 0f06 d7a1 ffff ffff        E..+...........
0x0010   aa81 29ab 7a69 0203 0000 0000 0000 0000        ..).zi..........
0x0020   5014 0000 8cc9 0000 636b 6f00 0000             P.......cko...
```

3.3.4 Probability of Source Address was spoofed

With total certainty we can say that the IP address has been spoofed. We reached this conclusion because the RFC 1122 (Internet hosts requirements – communication layers)[17], establishes in a very clear manner that the address 255.255.255.255 is an a limited broadcast address and cannot be used as a source address.
On the other hand, the port 31337 is a port used by Internet attacks as it can be seen in Internet Storm Center[18 – 19]

3.3.5 Attack Mechanism

The traffic associated with this attack sends frames with bits SYN ACK set, without a previous SYN being sent. This can be used as a evasion mechanism against Intrusion Detection Systems (IDSs). Additionally the ID of all packets is 0. The packets come from the internet with the source address 255.255.255.255, and as we described before, it is not a permitted address for traffic coming from the internet. The idea of this attack is to look for hosts that shown weaknesses in their configuration and respond to these altered packets. If these attacks were effective (there is no evidence in the logs) the destination addresses would respond to the broadcast address of the internal network generating possible problems of excessive traffic in the internal network, and also could generate a Denial of Service (in direct relationship with the amount of packets sent).

Another analysis on this Snort alert that indicates a Q attack, was presented by Les Gordon.[20].

3.3.6 Correlations

There is a publication of very high level written by Les Gordon which provides detailed information on the Trojan Q:
http://www.sans.org/resources/idfaq/qtrojan.php[19].
There is also a GCIA practical also written by Mr. Gordon that has already been referred to previously:
http://www.giac.org/practical/GCIA/Les_Gordon.doc[20].
We also made reference on a Meter Stone's GCIA practical that was graded with honors:
http://www.giac.org/practical/GCIA/Pete_Storm_GCIA.pdf.[21]
Internet Storm Center is an excellent reference when information about these ports is required. From there we take the following URL's:
(http://isc.sans.org/diary.php?date=2004-04-23)[18] and
http://www.sans.org/resources/idfaq/oddports.php[19]
Finally, there is no exact relationship with any entry inside the CVE, but the one that is most approximate, although in our opinion is too general, is the one referenced in the URL:
http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=1999-0660[22].

3.3.7 Evidence of Active Targeting

No existing evidence in all the information analyzed of Activity in the target hosts. It looks as if there was a random IP addresses selection and verifies if there was "any luck" in one of them. We think that if the attack would have been effective, it could have been observed with much more traffic in the internal network.

3.3.8 Severity

Severity = (criticality + lethality) – (system countermeasures + network countermeasures)

Severity = (2+4) – (4+1). Severity = 1

Criticality (2) there is not enough information to determine the importance of each of the target hosts, because we can not perceive any activity in them.

Lethality (4) if the attack would have been effective the damages could have been considerate at all internal network level, even allowing taking control of the target hosts. For this reason it is considered as a highly dangerous attack.

System Countermeasures (4) there is no existing evidence that any answer was produced at any level in the operating system against the analyzed attack. In this matter and even though we do not have enough information about patches and hardening, among others, it seems that the systems are protected.

Network Countermeasures (1) due to the fact that suspicious packets can access the internal network with spoofed addresses (which should have been eliminated at the perimeter) and the existing packets alterations, we can establish that the measures of defense inside the network against these types of attacks is very deficient.

4. Network Statistics

In this section we will show the five most "talkative" addresses inside the analysis, those who generate more volume of information, as well as the first five (5) ports or services that were used. Let's remember that the files used in this analysis were:

- 2002.10.14
- 2002.10.15
- 2002.10.16

The criteria of selection used were to take those IP that generated the most amounts of alerts during the monitored time.

4.1 Most Active Addresses

Source Address	Total Alerts	Type of Alerts
170.129.50.120	541	- P2P Outbound GNUTella client request - CHAT IRC nick change
64.125.138.190	146	SHELLCODE x86 inc ebx NOOP
255.255.255.255	79	BACKDOOR Q access
63.111.48.133	77	SHELLCODE x86 NOOP
211.47.255.20	45	BAD-TRAFFIC tcp port 0 traffic

1) 170.129.50.120 had a total of 527 alerts generated in two categories:
P2P Outbound GNUTella client request where clients were detected trying to connect to a GNUTella server to share files. This can bring as a consequence the downloading of the worm GNUTella by the clients as well as to give access to confidential information[23].
CHAT IRC nick change: this event is generated when CHAT activity is reported inside the network. This type of traffic can be harmful because of the kind of information it exchanges and downloads through this type of communications services[24].

2) 64.125.138.190 generated a total of 146 alerts with the message SHELLCODE x86 inc ebx NOOP. This alert is generated when there is an attempt of code execution at a shell level in a host of the internal network from a

external source address. An important information is that this alert can generate false positives when ports are not ignored each time a continuous chains of 24 characters “C” in a row appears. This is the case of the present alerts. The same correspond to false positives. We show you the following frame as an example:

```
11/15/2002  17:46:16.116507  0:3:e3:d9:26:c0  0:0:c:4:b2:33  0800  902:
64.125.138.190.80 > 170.129.50.120.62855: P [tcp sum ok] 3917272213:39172
73061(848) ack 38074492 win 33120 (DF) (ttl 49, id 32115, len 888)
0x0000   4500 0378 7d73 4000 3106 20d8 407d 8abe        E..x}s@.1...@}..
0x0010   aa81 3278 0050 f587 e97c d495 0244 f87c        ..2x.P...|...D.|
0x0020   5018 8160 3df5 0000 ffd8 ffe0 0010 4a46        P..`=.........JF
0x0030   4946 0001 0101 0048 0048 0000 ffdb 0043        IF.....H.H.....C
0x0040   000b 0708 0a08 070b 0a09 0a0c 0c0b 0d10        ................
0x0050   1b12 100f 0f10 2118 1914 1b27 2329 2927        ......!....'#))'
0x0060   2326 252c 313f 352c 2e3b 2f25 2636 4a37        #&%,1?5,.;/%&6J7
0x0070   3b41 4346 4746 2a34 4d52 4c44 523f 4546        ;ACFGF*4MRLDR?EF
0x0080   43ff db00 4301 0c0c 0c10 0e10 2012 1220        C...C...........
0x0090   432d 262d 4343 4343 4343 4343 4343 4343        C-&-CCCCCCCCCCCC
0x00a0   4343 4343 4343 4343 4343 4343 4343 4343        CCCCCCCCCCCCCCCC
0x00b0   4343 4343 4343 4343 4343 4343 4343 4343        CCCCCCCCCCCCCCCC
0x00c0   4343 4343 4343 ffc0 0011 0800 2900 3603        CCCCCC......).6.
0x00d0   0111 0002 1101 0311 01ff c400 1800 0003        ................
0x00e0   0101 0000 0000 0000 0000 0000 0000 0405        ................
0x00f0   0607 03ff c400 3310 0001 0401 0105 030a        ......3.........
```

3) 63.111.48.133 this IP address sent a total of 77 information packets that generated the alert SHELLCODE x86 NOOP, allowing the attacker to obtain benefits from the functions written in an unsecured way, placing him/her in the capacity of executing an arbitrary code[26.]

4) 255.255.255.255 was chosen because it was used as the source address of the external packets which is prohibited. This address is used for limited broadcast. By spoofing this IP address as source, 79 alerts were generated associated with the Backdoor Q.

5) 211.47.255.20 this address focused its activity in sending crafted packets of information in which the destination port 0 was established as well as ID 0 in each packet. This type of traffic should not be observed in normal conditions. This can be seen as an Operating System fingerprinting.

4.2 Services/Ports Most Used

Port/Services	Total Occurrences
80	4175
0	151
6667	150
515	76
63414	48

1) Port 80: generated 4175 alerts. This port is generally target of attacks because is used for Internet access as well as its a Web servers open port in which a never ending amount of vulnerabilities can be exploited. Due to the fact that it was determined that the IP 170.129.50.120 offers Web services, it is logical that this type of traffic be detected in an elevated number of packets.

2) Port 0: this is a reserved port and the sending of packets to it represents an alteration of the packet with the probability of recognition actions being attempted. Remember that an important number of alerts were generated on itself.

3) Port 6667: this is a CHAT port. The relevance of this type of traffic is represented by the generating of 150 alerts due to the fact that associated IRC frames use the word NICK; from which we can presume that some host has been compromised and is acting as a Zombie waiting for an order to trigger some type of attack. Only one NICK R00teD-04 shows up in the analyzed logs. The following is an example of what is mentioned. In bold we observe the word NICK which triggers the alarm:

```
11/14/2002   20:59:25.476507   0:0:c:4:b2:33   0:3:e3:d9:26:c0   0800   70:
170.129.50.120.61599 > 217.8.139.18.6667: P 2914527739:2914527755(16) ack
4150994086 win 15408 (DF)
0x0000   4500 0038 0595 4000 7b06 b916 aa81 3278        E..8..@.{.....2x
0x0010   d908 8b12 f09f 1a0b adb8 29fb f76b 24a6       ..........)..k$.
0x0020   5018 3c30 3523 0000 4e49 434b 2052 3030        P.<05#..NICK.R00
0x0030   7465 442d 3030 340a                            teD-004.
```

4) Port 515: shows a significant number of appearances, a total of 150. The alerts were generated by an attack of Backdoor Q that was mentioned in detail where there is an attempt to scan the host in search for information of interest to the attacker using the spoofed address 255.255.255.255.

5) Port 63414: all the traffic generated to this port comes from the address 129.118.2.10 and is associated with the attacker searchs in order to take advantage of written function in an unsecured way, allowing through the filling of addresses spaces of NOP code to later be able to execute arbitrary code. Doing a revision in the ARIN[28] we can see to whom this address belongs:

```
OrgName:    Texas Tech University
OrgID:      TTU-1
Address:    Telecommunications Department
Address:    2500 Broadway
City:       Lubbock
StateProv:  TX
PostalCode: 79409
Country:    US

NetRange:   129.118.0.0 - 129.118.255.255
CIDR:       129.118.0.0/16
NetName:    TTUNET
NetHandle:  NET-129-118-0-0-1
Parent:     NET-129-0-0-0-0
NetType:    Direct Assignment
NameServer: MINOTAUR.NET.TTU.EDU
NameServer: UNICORN.NET.TTU.EDU
NameServer: CHINATI.OTS.TTU.EDU
Comment:
RegDate:    1987-11-06
Updated:    2003-02-04

TechHandle: JS450-ARIN
TechName:   Stalcup, J
TechPhone:  +1-806-742-3698
TechEmail:  J.Stalcup@ttu.edu

OrgTechHandle: NOC1620-ARIN
OrgTechName:   Network Operations Center
OrgTechPhone:  +1-806-742-4858
OrgTechEmail:  noc@ttu.edu
```

4.3 Most Dangerous External Source Addresses

1) 211.47.255.20: this IP address sent in a continuous manner SYN packets using port 0, that is a reserved port. The using of this port, shows that the packets have been crafted manually trying to achieve operating system recognition function of the target addresses. Searching the address in APNIC[27], we have that the address belongs to:

```
inetnum:     211.46.0.0 - 211.49.255.255
netname:     KRNIC-KR
descr:       KRNIC
descr:       Korea Network Information Center
country:     KR
admin-c:     HM127-AP
tech-c:      HM127-AP
remarks:     *******************************************
remarks:     KRNIC is the National Internet Registry
remarks:     in Korea under APNIC. If you would like to
remarks:     find assignment information in detail
remarks:     please refer to the KRNIC Whois DB
remarks:     http://whois.nic.or.kr/english/index.html
```

```
remarks:     *****************************************
mnt-by:      APNIC-HM
mnt-lower:    MNT-KRNIC-AP
changed:     hm-changed@apnic.net 19991118
changed:     hm-changed@apnic.net 20010606
changed:     hm-changed@apnic.net 20040623
status:      ALLOCATED PORTABLE
source:      APNIC
person:      Host Master
address:     11F, KTF B/D, 1321-11, Seocho2-Dong, Seocho-Gu,
address:     Seoul, Korea, 137-857
country:     KR
phone:       +82-2-2186-4500
fax-no:      +82-2-2186-4496
e-mail:      hostmaster@nic.or.kr
nic-hdl:     HM127-AP
mnt-by:      MNT-KRNIC-AP
changed:     hostmaster@nic.or.kr 20020507
source:      APNIC
```

2) 255.255.255.255: this address is suspicious because it was spoofed to be used as external when this is not allowed. It also used port 31337 as a destination port that is a hackers known port.

3) 63.111.48.133 this address generated multiple attempts to search functions developed in an inappropriate way that would allow exploiting vulnerabilities by the execution of arbitrary code. Investigating in ARIN[28] we can determine that the IP belongs to:

Sybari Software, Inc. UU-63-111-48-128 (NET-63-111-48-128-1)
63.111.48.128 - 63.111.48.159

5. CORRELATIONS WITH PREVIOUS GCIA CERTIFICATIONS

In this section we will make reference to those GCIA practical assignments that served as support for the construction of this document. According to the normative established in this assignment, the documents selected to be consulted should be those whose number of certification is above 600

5.1 Detect 1: Code Red
http://www.giac.org/practical/GCIA/Josh_Berry_GCIA.pdf[4]
http://www.giac.org/practical/GCIA/Mark_Faske_GCIA.pdf[29]
http://www.giac.org/practical/GCIA/John_Petkovsek_GCIA.pdf[30]

5.2 Detect 2: TCP Traffic to port 0 with ID 0
http://www.giac.org/practical/GCIA/Eric_Evans_GCIA.pdf[15]
http://www.giac.org/practical/GCIA/Tyler_Hudak_GCIA.pdf[31]
http://www.giac.org/practical/GCIA/Joe_Bowling_GCIA.pdf[10]

5.3 Detect 3: Backdoor Q
http://www.giac.org/practical/GCIA/Tyler_Hudak_GCIA.pdf[31]
http://www.giac.org/practical/GCIA/Greg_Bassett_GCIA.pdf[32]
http://www.giac.org/practical/GCIA/Rob_McBee_GCIA.pdf[33]

6. OTHER TYPES OF MALICIOUS ACTIVITIES

Inside the analyzed information logs, there is no detection of any other malicious activity different to the recognition activities, attempt to access Proxy services, manual packets alterations (DF and MF bits active simultaneously), traffic with same source/destination ports and traffic directed to port 0 besides the Code Red contained in the case of the DF/MF bits attack. There are several aspects that should be improved and will be dealt with in the recommendations section.

7. DEFENSIVE RECOMMENDATIONS

After finish the analysis, we came to the conclusion that the university must make a mayor efforts on perimeter protection.

In the case of Code Red, the best recommendation is to apply the appropriate patch for the IIS servers, by which the action of the worm will be avoided. The patch is found in the following URL:
http://www.microsoft.com/technet/security/bulletin/MS01-033.mspx[6].
This patch is exclusive for Code Red.

Regarding the necessary defensive measures to handle the packets directed to the port 0, as well as the attack of Backdoor Q that uses as source address 255.255.255.255 and destination port 31337, the best recommendation is the use of firewalls. Seeing that the fields of source/destination address as well as the destination port fields are located in the IP header, it can be analyzed by the firewall, done differently when the attack is immersed in the payload of the packet, situation in which the firewall will not analyze this frame portion and allows it to continue to and from the internal network. In this matter the firewalls will be enough to avoid these types of incidents. When considering valid the fact that a firewall exists in the network topology design as seen in figure 4, we allow ourselves to conclude that in this firewall the packet filtering rules are not well defined. To search for measures still more effective and thinking on new security strategies that is directed for prevention, we recommend to implement multifunctional appliances solutions that not only provide a firewall they contain Intrusion Prevention (IPS) and antivirus gateway modules amongst others.

Even though it was not the case of this analysis but trying to enriching results for the university, we make some additional recommendations:

- In the case of constant scanning in search of available proxy services, the recommendation is to use reverse proxys, as well as strong mechanisms of user authentification.
- Give constant follow ups on the logs generated by the tools of security. Remember that the attackers are addicted to read/analyze logs in order to know our weaknesses.
- Carry out vulnerability analysis periodically over the perimeter as well as internal that allow us to meet with time in advance which are the existing gaps inside the network and the corrective measures that should be applied
- Use intrusion prevention tools in the network particularly in the perimeter segments
- In the case of the first attack, we recommend to apply a patch was recommended to solve the problem. Nevertheless this solution is palliative, updates are always necessary and everyday manufacturers release patches or hot fixes with more celerity by which the manual application of them is practically impossible. This is why we recommend the adoption of automatic patching solutions that can work in combination with the vulnerability assessment, being the outcome of this last one the entry of this patching system and have total automatic control of everything related to vulnerabilities management.

PART III. ANALYSIS PROCESS

In this section we will describe the platform used and the steps followed in the analysis. We will describe the tools used as well as the inconveniences arised and the way to fix them.

For the development of this investigation the following hardware and software elements were used.

✓ Laptop Pentrium IV, 512MB RAM, Windows XP Professional SP2
✓ Snort Version 2.0.4-ODBC-MySQL-FlexRESP-WIN32 (Build 97) with a configuration file snort.conf, v 1.124 16-05-2003.
✓ Snortsnarf version 2.1111.1 adding the julianday.pm, timezone.pm and parsedate.pm modules
✓ Windump 3.6.2
✓ Winpcap 2.3
✓ Perl 5.6.1 (neccesary to use Snortsnarf)
✓ Ethereal 0.10.5a.
✓ Textpad 4.7.3

On Snort we enabled all the backdoor rules that are initially set as a comment to try to have the most exhaustive and precise analysis. To our knowledge the following rules were enabled:

```
include $RULE_PATH/web-attacks.rules
include $RULE_PATH/backdoor.rules
include $RULE_PATH/shellcode.rules
include $RULE_PATH/policy.rules
include $RULE_PATH/porn.rules
include $RULE_PATH/info.rules
include $RULE_PATH/icmp-info.rules
include $RULE_PATH/virus.rules
include $RULE_PATH/chat.rules
include $RULE_PATH/multimedia.rules
include $RULE_PATH/p2p.rules
```

In addition , the preprocessors stream4_reassemble and stream4: detect_scans, disable_evasion_alerts were disabled by putting a # in front of each one of them. This was to avoid the trace of connections because the logs only contain packets of information with problems, this would cause lose of information in the processing of the logs. The alert.ids files generated were visualized with Textpad and the graphics were generated with Snortsnarf giving these *.ids files as entries.
In order to Snortsnarf worked properly the modules julianday.pm, timezone.pm and parsedate.pm, were downloaded and installed in the Perl’s Time folder.

At the time of accomplishing the packets analysis was necessary to disable the antivirus installed in the laptop (Symantec Antivirus Corporate Edition), because when we tried to open the files for the analysis (particularly 2002.10.14) the antivirus would detect the virus and eliminate the file.

REFERENCES

[1] Perdue, Robert. “GCIA Practical”. Septiembre 29, 2004
URL: http://www.giac.org/practical/GCIA/Rob_Perdue_GCIA.pdf

[2] Security Incidents: Initial analysis of the .ida "Code Red" Worm
URL: http://seclists.org/lists/incidents/2001/Jul/0083.html

[3] ISS X-Force Response to Concern About the "Code Red" Worm
URL: http://xforce.iss.net/xforce/alerts/id/advise89

[4] Berry, Josh. “GCIA Practical”. Agosto 18, 2004
URL: http://www.giac.org/practical/GCIA/Josh_Berry_GCIA.pdf

[5] Trend Micro. “CodeRed.A”. Julio 30, 2001
URL:
http://www.trendmicro.com/vinfo/virusencyclo/default5.asp?VName=CODERED.A&VSect=T

[6] Microsoft Security Bulletin MS01-033. “Unchecked Buffer in Index Server ISAPI Extension Could Enable Web Server Compromise”
URL: http://www.microsoft.com/technet/security/bulletin/MS01-033.mspx

[7] Base de Datos Mitre CVE. “Buffer overflow in ISAPI extension (idq.dll)”
URL: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2001-0500

[8] eEye Digital Security. “Microsoft Internet Information Services Remote Buffer Overflow (SYSTEM Level Access)”. Junio 18, 2001
URL: http://www.eeye.com/html/Research/Advisories/AD20010618.html

[9] CERT Advisory CA-2001-19. “"Code Red" Worm Exploiting Buffer Overflow In IIS Indexing Service DLL”
URL: http://www.cert.org/advisories/CA-2001-19.html

[10] Bowling, Joe.”GCIA Practical” Abril 2003
URL: http://www.giac.org/practical/GCIA/Joe_Bowling_GCIA.pdf

[11] Port 0 OS Fingerprinting
URL: http://www.securiteam.com/securityreviews/5XP0Q2AAKS.html

[12] Base de Datos RFC. “RFC 1700: Assigned Numbers”
URL: http://www.faqs.org/rfcs/rfc1700.html

[13] Port 0 OS Fingerprinting by Ste Jones NetworkPenetration.com
URL: http://securityfocus.com/archive/129/330750

[14] Material Curso Intrusion Detection In-Depth. SANS (acceso solo a estudiantes activos)
URL: http://giactc.giac.org/cgi-bin/momgate IDs: ID_23_1203.pdf, ID_25_1203.pdf, ID_42_1203

[15] Evans, Eric.”GCIA Practical” Diciembre 11, 2003
URL: http://www.giac.org/practical/GCIA/Eric_Evans_GCIA.pdf

[16] Snort Signature Database. “Backdoor Q Access”
URL: http://www.snort.org/snort-db/sid.html?id=184

[17] Base de Datos RFC. “Requirements for Internet Hosts - Communication Layers”
URL: http://www.faqs.org/rfcs/rfc1122.html

[18] Internet Store Center. “Move to Yellow, Potential PCT worm, No Osama has NOT been captured, New Virus, Symantec Firewall Vulnerability”
URL: http://isc.sans.org/diary.php?date=2004-04-23

[19] SANS, Intrusion Detection FAQ. “What port numbers do well-known trojan horses use?”
URL: http://www.sans.org/resources/idfaq/oddports.php

[20] Gordon, Les.”GCIA Practical” Noviembre 22, 2002
URL: http://www.giac.org/practical/GCIA/Les_Gordon.doc

[21] Storm, Pete “GCIA Practical” Noviembre 15, 2003
URL: http://www.giac.org/practical/GCIA/Pete_Storm_GCIA.pdf

[22] Base de Datos Mitre CVE. CAN-1999-0660
URL: http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=1999-0660

[23] Snort Signature Database. “P2P Outbound GNUTella client request”
URL: http://www.snort.org/snort-db/sid.html?id=556

[24] Snort Signature Database. “CHAT IRC nick change”
URL: http://www.snort.org/snort-db/sid.html?id=542

[25] Snort Signature Database. “SHELLCODE x86 inc ebx NOOP”
URL: http://www.snort.org/snort-db/sid.html?id=1390

[26] Snort Signature Database. “SHELLCODE x86 NOOP”
URL: http://www.snort.org/snort-db/sid.html?id=648

[27] Base de Datos APNIC. Entradas Whois
URL: http://www.apnic.net/apnic-bin/whois.pl

[28] Base de Datos ARIN. Entradas Whois
URL: http://ws.arin.net/cgi-bin/whois.pl

[29] Faske, Mark."GCIA Practical" Diciembre 7, 2003
URL: http://www.giac.org/practical/GCIA/Mark_Faske_GCIA.pdf

[30] Petkovsek."GCIA Practical" Mayo 30, 2003
URL: http://www.giac.org/practical/GCIA/John_Petkovsek_GCIA.pdf

[31] Judak, Tyler. "GCIA Practical"
URL: http://www.giac.org/practical/GCIA/Tyler_Hudak_GCIA.pdf

[32] Basset, Greg. "GCIA Practical" Septiembre 21, 2003
URL: http://www.giac.org/practical/GCIA/Greg_Bassett_GCIA.pdf

[33] McBee, Rob. "GCIA Practical" Julio 8, 2003
URL: http://www.giac.org/practical/GCIA/Rob_McBee_GCIA.pdf

[34] Security Focus. "Examining a Public Exploit, Part I"
URL: http://www.securityfocus.com/infocus/1795

[35] Security Focus. "Examining a Public Exploit, Part II"
URL: http://www.securityfocus.com/infocus/1801

[36] The Honeynet Project. Know Your Enemy. Unknow: Addison-Wesley. 2002.

[37] Northcutt, Stephen; Zeltser, Lenny; winters, Scott; Frederick, Karen Ken; Ritchey, Ronald. Inside Network Perimeter Security. Unknow: New Riders. 2003.

[38] Northcutt, Stephen; Novak Judy. Network Intrusion Detection. An Analyst's Handbook. Unknow: New Riders. 2000.