Introduction to WMIC

It’s quite possible you’ve never heard of the Windows Management Instrumentation Command-line (WMIC), but this well kept secret command-line tool is immensely powerful for gathering information from Windows-based systems. Because it can be used both locally and over the network and is installed by default on most Windows-based systems since Windows 2000, it’s exceedingly useful for both penetration testing and forensics tasks.

What is WMIC?

If you’ve done any scripting for the Windows platform, you’ve probably bumped into the Windows Management Instrumentation (WMI) scripting API, which can be used to enumerate all kinds of information. The WMIC command-line tool is basically another front-end to access the WMI framework, with the added bonus that numerous queries are pre-defined. The pre-defined queries mean that you won’t necessarily need to spend any time learning the WMI Query Language (WQL), which is syntactically similar to SQL.

WMIC is included in the default installation of Windows XP (excluding Home edition) and Windows Server 2003. Although WMIC is not included on Windows 2000, you can still use a Windows XP or Server 2003 client to remotely query Windows 2000 systems and receive similar results. The first time you run WMIC you'll see a message that WMIC is being installed, but no media is required for installation, nor will anything appear in the Add/Remove Programs list.

Basic WMIC Usage

There are two modes of usage for WMIC, from the command line directly and from within its own shell, similar to nslookup. As the features are nearly identical, we won’t cover the WMIC shell in this article. However, note that the interactive version does include a failsafe mode that asks for confirmation before modifying or deleting objects, but it isn’t on by default.

Most WMIC commands are issued in the following format:

```
wmic [credentials] [area] [querystring]
```

For example, you can collect a list of groups on the local system using the following command:

```
wmic group list brief
```

which will return output similar to this:

```
Caption Domain Name SID
Lab7\Administrators Lab7 Administrators S-1-5-32-544
Lab7\Backup Operators Lab7 Backup Operators S-1-5-32-551
Lab7\Guests Lab7 Guests S-1-5-32-546
Lab7\Network Configuration Operators Lab7 Network Configuration Operators S-1-5-32-556
Lab7\Power Users Lab7 Power Users S-1-5-32-547
Lab7\Remote Desktop Users Lab7 Remote Desktop Users S-1-5-32-555
Lab7\Replicator Lab7 Replicator S-1-5-32-552
Lab7\Users Lab7 Users S-1-5-32-545
Lab7\Debugger Users Lab7 Debugger Users S-1-5-32-545

You can also perform the same data collection over the network without ever logging into the remote machine provided you know have some administrative credentials that the remote system will accept. The same command issued against a remote system in another domain looks like this:

```
wmic /user:"FOREIGN DOMAIN\Admin" /password:"Password" /node:192.168.33.25 group list brief
```

```
Caption Domain Name SID
REMOTE-DESK\Administrators REMOTE-DESK Administrators S-1-5-32-544
REMOTE-DESK\Backup Operators REMOTE-DESK Backup Operators S-1-5-32-551
REMOTE-DESK\Guests REMOTE-DESK Guests S-1-5-32-546
REMOTE-DESK\Network Configuration Operators REMOTE-DESK Network Configuration Operators S-1-5-32-556
REMOTE-DESK\Power Users REMOTE-DESK Power Users S-1-5-32-547
REMOTE-DESK\Remote Desktop Users REMOTE-DESK Remote Desktop Users S-1-5-32-555
REMOTE-DESK\Replicator REMOTE-DESK Replicator S-1-5-32-552
REMOTE-DESK\Users REMOTE-DESK Users S-1-5-32-545
REMOTE-DESK\Debugger Users REMOTE-DESK Debugger Users S-1-5-32-545
FOREIGN_DOMAIN\HelpServicesGroup FOREIGN_DOMAIN HelpServicesGroup S-1-5-21-789336098-1078081533-389522115-1001
FOREIGN_DOMAIN\__vmware__ FOREIGN_DOMAIN__vmware_ S-1-5-21-789336098-1078081533-389522115-1004
FOREIGN_DOMAIN\Cert Publishers FOREIGN_DOMAIN Cert Publishers S-1-5-21-1948120765-2568877423-583383040-517
FOREIGN_DOMAIN\IAS Servers FOREIGN_DOMAIN IAS Servers S-1-5-21-1948120765-2568877423-583383040-553
FOREIGN_DOMAIN\HelpServicesGroup FOREIGN_DOMAIN HelpServicesGroup S-1-5-21-1948120765-2568877423-583383040-1000
FOREIGN_DOMAIN\TeleClient FOREIGN_DOMAIN TeleClient S-1-5-21-1948120765-2568877423-583383040-1002
FOREIGN_DOMAIN\UnsaldpProxy FOREIGN_DOMAIN UnsaldpProxy S-1-5-21-1948120765-2568877423-583383040-1117
FOREIGN_DOMAIN\Domain Admins FOREIGN_DOMAIN Domain Admins S-1-5-21-1948120765-2568877423-583383040-1118
FOREIGN_DOMAIN\Domain Admins FOREIGN_DOMAIN Domain Admins S-1-5-21-1948120765-2568877423-583383040-1119
FOREIGN_DOMAIN\Domain Computers FOREIGN_DOMAIN Domain Computers S-1-5-21-1948120765-2568877423-583383040-1115
FOREIGN_DOMAIN\Domain Controllers FOREIGN_DOMAIN Domain Controllers S-1-5-21-1948120765-2568877423-583383040-1116
FOREIGN_DOMAIN\Domain Users FOREIGN_DOMAIN Domain Users S-1-5-21-1948120765-2568877423-583383040-1117
FOREIGN_DOMAIN\Enterprise Admins FOREIGN_DOMAIN Enterprise Admins S-1-5-21-1948120765-2568877423-583383040-1112
FOREIGN_DOMAIN\Group Policy Owner Owners FOREIGN_DOMAIN Group Policy Owner Owners S-1-5-21-1948120765-2568877423-583383040-1113
FOREIGN_DOMAIN\Schema Admins FOREIGN_DOMAIN Schema Admins S-1-5-21-1948120765-2568877423-583383040-1114
FOREIGN_DOMAIN\Shared FOREIGN_DOMAIN Shared S-1-5-21-1948120765-2568877423-583383040-1115
```

Note that you can issue ANY of the of the WMIC commands over the network in this fashion as a means of gathering information about the host. Now that we’ve seen the basics, let’s move to specific applications.

WMIC in Vulnerability and Penetration Testing

In vulnerability and penetration testing, system footprinting is key. The more information that can be collected about a specific system or group of systems, the greater the likelihood that those systems can be compromised.

Granted, using WMIC requires administrative access on the remote host, but since most IT departments maintain standard images for each collection or group of workstations and servers, information you can obtain from one host is likely to be applicable to other similar systems. Furthermore, for default configurations of the event log and auditing processes, WMIC requests won’t be logged, so all of your enumerations can be undertaken in stealth mode.

The following are examples of useful information we can collect through WMIC:

Process Listings

WMIC can collect a list of the currently running processes similar to what you’d see in “Task Manager” using the following command:

```
wmic process list
```

Note that some of the WMIC built-ins can also be used in “brief” mode to display a less verbose output. The process built-in is one of these, so you could collect more refined output using the command:

```
wmic process list brief
```

About half of the pre-defined WMIC queries that I’ve used seem to have a brief version available, but I use the full versions almost exclusively.
Environment Settings
You can collect a listing of the environment variables (including the PATH) with this command:

wmic environment list

User and Groups
Local user and group information can be obtained using these commands:

wmic useraccount list
wmic group list
wmic sysaccount list

For domain controllers, this should provide a listing of all user accounts and groups in the domain. The “sysaccount” version provides you with system accounts built-in and otherwise, which is useful for any extra accounts that may have been added by rootkits.

Patch Management
Need to know if there are any missing patches on the system? WMIC can help you find out with this command:

wmic qfe list

The QFE here stands for “Quick Fix Engineering”. The results also include the dates of install should that be needed from an auditing standpoint.

Shares
Enumeration of all of the local shares can be collected using the command:

wmic share list

The result will also include hidden shares (named with a $ at the end).

Network Adapters
Looking for dual-homed systems to find other networks? WMIC can assist you! Use the following command to extract a list of network adapters and IP address information.

wmic nicconfig list

Services
WMIC can list all of the installed services and their configurations using this command:

wmic services list

The output will include the full command used for starting the service and its verbose description.

Of course, these are just samplings of the dozens of predefined aliases within WMIC. You can also go beyond the predefined aliases using WQL queries to collect and set any of the many thousands of parameters accessible through WMI.

WMIC in Forensics
In forensics, it’s often important to get as much information about the running system as possible before the system can be shut down. You’d also like to collect that information while keeping close records that account for your own actions and leave the smallest footprint possible on the system. Though WMIC wasn’t really designed with this in mind, it certainly works. Since WMIC is included by default on most Windows systems and can be executed remotely, that makes it all the more desirable.

Many of the built-in aliases already described are also useful in forensics, but there are a few others not yet mentioned. These can be executed in the same fashion and include the following:

- **Job** - Accesses the local jobs queued using the scheduler service.
- **RecoverOS** - Find out where the memory dumps are stored in the event of an emergency shutdown.
- **Startup** - Identifies many of the processes set to launch at system start-up.

Another interesting feature of WMIC is its ability to record the run-time command executed and runtime configuration all in one XML file. A recorded session might look something like this:

wmic /record:users_list.xml useraccount list

Of course, since WMIC wasn’t designed as a recording device, there are some caveats to using the XML. First, you can only use XML output, there are no other formats defined. Additionally, you need to specify a new filename for each command. If the file already exists it will be silently overwritten, which is obviously undesirable.

Other WMIC Capabilities
We’ve only scratched the surface here with what WMIC can do. Although these examples have only shown you how to collect data from WMIC, you can also modify most of the parameters that we’ve collected, as well as kill existing processes, start new processes detached from the console and much more! I encourage you to read the command line help available for WMIC and search out additional information.

William Lynch is a manager with CTG’s Information Security Services Practice.