Penetration Testing: Stopping an Unstoppable Windows Service

Every penetration tester has a toolkit they use for escalating their privileges on the network. In some cases, the tester will copy the toolkit over to a target system once it has been compromised. However, anti-virus software has gotten pretty good at catching tools commonly included in such toolkits. To get around this problem, many penetration testers simply disable the anti-virus services prior to copying over their toolkit. Usually, disabling Windows services is trivial, but some don't shutdown as easy as we would like. This blog describes a relatively simple way to stop those resistant services.

Windows Services

Before we jump into it, let's talk a little bit about Windows Services. Windows Services are applications that run quietly in the background and perform tasks that don't usually require any user interaction (like anti-virus software). Each service is configured with a number of different settings that control how it operates. Service configurations could easily take up their own blog entry, but for the purpose of this blog I will be focusing on the "Startup Type" and "Service Type" settings. Being the intuitive readers that you are, you may have already guessed that the "Startup Type" dictates how each service is started. Below are the startup types used by most Windows services:

Startup Types	Description
Automatic	If a service startup type is set to automatic it will automatically start each and every time the computer is restarted. It will start the service even if a user is not logged on.
Manual	If a service startup type is set to Manual it must be manually started by the user or application.
Disabled	If a service is Disabled it cannot be started until its start type has been changed to another option.

Each Windows service is also assigned a Service Type, which specifies how each service runs on the system. I've listed the two most common Windows service types below:

Service Type	Description
Win32_OWN_PROCESS	Win32 services that run as a stand-alone process.
Win32_SHARE_PROCESS	Win32 services that are capable of sharing address space with other services of the same type.

Additional configurations can also be set on the Service Type that determines how users and applications are allowed to control the service. Some of those configurations include:

- NOT_STOPPABLE and STOPPABLE

- NOT_PAUSABLE and PAUSABLE
- IGNORES_SHUTDOWN and ACCEPTS_SHUTDOWN

If you’re wondering how the services are configured on your system issue the following command from the command line.

```
SC QUERY
```

```
SC STOP
```

However, if a service is running as under its own process and configured as NOT_STOPPABLE it doesn’t always go down quite so easily. The standard method used to stop that type of service is to set its “Startup Type” to “Disabled” and then restart the system. That may be an option for a systems administrator, but it’s typically considered bad form to shutdown, or restart a system during a penetration test.**Stopping Windows Services** Most of the services in Windows are configured to startup automatically as part of a shared process. Typically those services are also configured with the STOPPABLE, PAUSABLE, and ACCEPTS_SHUTDOWN settings so they can be managed by users and other applications. Those services can be easily stopped using the Services Management Console, the “SC” command, or the “NET STOP” command. For example:

Stopping NOT_STOPPABLE Windows Services

Since I’m guessing most of you don’t want to restart the target server I am happy to provide an alternative. Corny as it may be, stopping a NOT_STOPPABLE service is as easy as 1-2-3.

1. First set the Startup Type to “Disabled”. This will prevent the service from restarting once it has been stopped. This can be done via the Services Management Console, “SC” command, or the registry. Just for fun I’ve provided an example of how to disable a service via the registry using the reg.exe application that comes with Windows.

   ```
   reg add HKEY_LOCAL_MACHINESYSTEMCurrentControlSetServices /V start /T
   reg_dword /D 4 /F
   ```

   For reference, I’ve also provided the other relative states and their associated registry values.

Registry Value	Startup Method
2	Automatic
3	Manual
4	Disabled

   However, if you want to take the easy route you can also set the service state to “Disabled” via the following “SC” command:

   ```
   SC CONFIG start= disabled
   ```

2. Next identify the executable that the target service is running. Use the “SC” command to get the executable:

   ```
   SC qc
   ```

You should be able to see it running as an active task using the TASKLIST command:

```
TASKLIST | FIND /I “”
```

3. Finally use TASKKILL to terminate the process using the known executable:

```
TASKKILL /F /IM ""
```

Please keep in mind that you may have to be running as administrator or LocalSystem to stop the target service. There are a number of ways to get LocalSystem on a Windows system, but that’s a story for another day. However, if you’re interested in further reading on the subject I’ve provided links to some of my previous blogs in the references section below.

Conclusion

Killing some services can be be a pain, but the next time you run across a service that doesn’t want to stop for you, remember that you have options that don’t include restarting the server.

References

- http://technet.microsoft.com/en-us/library/cc736564(WS.10).aspx
- http://technet.microsoft.com/en-us/library/cc785922(WS.10).aspx
- http://support.microsoft.com/kb/251192
- http://en.wikipedia.org/wiki/Windows_service
- http://netspi.com/blog/2009/10/05/windows-privilege-escalation-part-2-domain-admin-privileges/
- http://netspi.com/blog/2009/10/05/windows-privilege-escalation-part-1-local-administrator-privileges/