Find vulnerabilities with Metasploit

Develop exploits and find vulnerabilities in your enterprise's security

Sean-Philip Oriyano
August 18, 2009

Metasploit is a vulnerability scanning and exploit development tool that you can use to greatly enhance the security in your enterprise. Through the use of Metasploit, an organization can locate previously unknown weaknesses and begin addressing them.

When deploying a piece of software such as an operating system, application, or Web server, one of the biggest concerns is security. Is that application secure, or is there a hole or vulnerability that you don't know about? Most administrators interested in knowing the answer to this question simply rely on the various security bulletins posted on vendors' Web sites to obtain the information they need. However, if you're one of the many administrators and security professionals who wants to get deeper into his or her applications, you can actively verify whether your software is actually affected by a threat.

In the security field, several tools are designed to do what's known as vulnerability testing, each offering different features and benefits. Some of the more common tools include Nessus, SAINT, Nmap, and Metasploit—the focus of this article (see Related topics).

This article introduces at a basic level the purpose of Metasploit and the potential this software has in the enterprise. Although using Metasploit to perform your own vulnerability scanning is outside the scope of this article, many tutorials and books are available to help (see Related topics).

Vulnerability scanning

So, what's the purpose of vulnerability scanning, and how does Metasploit fit in? A vulnerability scanner is similar to other types of scanners—for example, port scanners help secure the network and the systems on it. The goal of such scans is to detect any weaknesses and use the results to address the problems before the "bad guys" do.

Common problems detected during the scanning process include locating buffer overflows, unpatched software, Structured Query Language (SQL) problems, and so on. How much a scanner detects depends on the software itself: Some scanners are much more aggressive in their scans, detecting missing patches or configuration errors, while others just seem to point you in the right direction.
Metasploit goes a step beyond regular vulnerability scanners: It provides you the ability to develop your own exploits and delivery mechanisms. The idea is that where other scanning methods rely on known problems, Metasploit allows you to develop your own with the intention of allowing more flexibility in scanning as well as research.

What's a vulnerability scanner?

Simply put, a vulnerability scanner is a piece of software specifically designed to search for and reveal weaknesses in an application or operating system. Each vulnerability scanner may perform its own scans and probes differently, but the result is more or less the same: a report or some sort of output informing you of the problems discovered.

The process of vulnerability scanning can be broken down into a series of steps, each of which is a vital task needed to locate the weaknesses in an application or operating system:

1. Provide the scanning software with the IP or host names of active systems. In this way, the scanner will be able to find open ports, services that respond, and any applications that may be currently running on the system.
2. The scanner generates a report informing you of what it discovered. This information may not be fully actionable and may require more research if the scanner ends at this step. With other scanners, this step may simply reveal the ports, services, and applications running so that the software can proceed to the next step.
3. The software runs probes against the ports and services discovered in Step 2, with the goal of returning information on the patch level, software revisions, or any one of a number of potential flaws. Modern vulnerability scanners have plug-ins designed to look for new weaknesses as they emerge. So, what a scanner does not find today it may very well find tomorrow after you perform an update.
4. Depending on how advanced the scanner is, the software may attempt to exploit vulnerabilities (if this feature is supported). In fact, some more modern scanners can have their aggressiveness at this step configured, meaning that they will not be so aggressive in their scanning that they bring down a system (which they can and have been known to do).

What is Metasploit?

The Metasploit Project is a series of projects that have spawned tools used for everything from defeating forensic methods and evading detection to its best-known software, the Metasploit Framework. Metasploit is not intended to be a hacking tool, even though it may sound like one on the surface. Indeed, the tool's primary goal is security research. However, as with any tool of this type, how it's used depends on the user. Remember, "With great power comes great responsibility," so be careful.

Metasploit had its genesis back in 2003, when it was based on the Perl scripting language. Since 2003, it has been rewritten from the ground up to use the Ruby programming language. Over the past couple of years or so, Metasploit has become one of the favored tools in the security research community, single-handedly responsible for creating some of the more sophisticated attacks against software and systems. In the right hands, this tool can offer a very powerful means of uncovering security vulnerabilities in software and assisting in their repair.
Currently, Metasploit runs on most of the Nix-based systems as well as on some of the other common operating systems in use. As of this writing, the stable version of Metasploit is Version 3.1.

**How Metasploit works**

Metasploit breaks down the steps mentioned earlier in the description of vulnerability scanners, with the goal of uncovering vulnerabilities. Basically, the scanner:

- Selects and configures the exploit to be targeted. This is the code that will be targeted toward a system with the intention of taking advantage of a defect in the software. **Note:** The pool of exploits is specific to an operating system. This pool changes by version of the system and grows all the time. Currently, Metasploit contains more than 400 exploits for most of the modern operating systems.
- Validates the exploit against the system with the intention of discovering whether the system is vulnerable to the exploit.
- Selects and configures a payload that will be used. This payload represents the code that will be run on a system after the scanner discovers an entry point into the system.
- Selects and configures the encoding and delivery technique that will be used. The goal of this step is to format the payload in such a way that it can evade entrenched intrusion-detection systems (IDSs).
- Executes the exploit.

**Metasploit in action**

This section provides an overview of what the Metasploit Framework does for users trying to discover vulnerabilities. I don't get into the specifics of how to run each exploit or write shell code. Instead, I give a broad overview of the process so you can get an idea of what's involved and determine whether the software is right for your testing needs.

For those of you who have never seen or worked with Metasploit, you will probably discover that the Metasploit Framework is surprisingly easy to use. In fact, you may find the product so easy to use that you may pay a little more attention to your patch-management process and stay informed of the latest fixes and vulnerabilities lest someone else point this tool your way.

When you think about it, this tool is the equivalent of an offensive weapon in the real world. The tool can be freely downloaded and installed with a myriad of prepackaged exploits that are already verified to work against specific platforms and applications. These exploits even come with their own matching payloads to make the process that much easier. Each preconfigured exploit is already set up to determine what to do upon delivery when the exploit is executed. **Note:** Even though Metasploit ships with several preconfigured exploits, it's important to realize that the software was envisioned as an exploit-development environment. You can use the tools that the software provides to test whether systems are vulnerable and how they will react when you direct specific payloads toward them.
Running an exploit

First, let's look at the process of running Metasploit. This process should apply to running most exploits and payload deliveries. In this scenario, I look generically at how you could launch an attack from one system against another.

**Note:** The following steps have been sanitized to prevent anyone from using them to cause mischief.

You can perform the steps here with just about any form of Linux® or other operating system from the command line or shell. You may have to look up specifics for your environment, but this process should give you an understanding of what's going on.

To use Metasploit to run an exploit, perform the following steps:

1. Download WHAX 3.0 for Linux (see [Related topics](#)).
2. From the command line, type the following command to launch the Metasploit console:
   ```bash
   # " ./msfconsole 
   ```
3. Choose an exploit to use against your target system. Three commands are available for this purpose:
   - **use:** Specify an exploit.
   - **show:** Display information in context.
   - **info:** Provide details about a specific module.
   ```bash
   <!--[if !supportLists]-->use
   <!--[if !supportLists]-->show
   <!--[if !supportLists]-->info
   <!--[endif]-->
   ```
   The format of the command you use to run an exploit is:
   ```bash
   "use <exploit name>"
   ```
4. Type the following command to start the exploit:
   ```bash
   use iis50_webdav_ntdll
   ```

Configuring the exploit

After you've selected your exploit, you need to configure how it will run. Specifically, you must specify the destination IP address and port against which to run the exploit. The **set** command allows you to specify configuration options; if you include the **show advanced** command, you'll be shown any options you can configure.

Use the following command to set options:

```bash
set RHOST <your test machine's IP address>
```

Then, press the Enter key. Next, type this code on the following line:

```bash
set RPORT 80
```

To perform a check to see whether the exploit functioned, type:
check

The results you get depend on the target.

If your check fails, you may need to configure other options for the target system—for example, details of the operating system.

Next, type:

```
show targets
```

Depending on the exploit, you may see additional information regarding the exploit, such as which services support it.

**The perfect attack**

You insert the Metasploit payload—also known as shell code—directly into the buffer that you're overflowing. In most cases, the payload is very specific, and which one you choose depends on the operating system and architecture of your target.

**Conclusion**

This article provided a high-level introduction to using Metasploit to provide a generic overview of your system's vulnerabilities. With a bit of work and research, you can develop your own exploits. After developing exploits, you can determine whether your applications and systems are vulnerable to exploits such as buffer overflows and holes such as SQL injections.
Related topics

- The **Nessus** vulnerability scanner provides several important scanning features, such as high-speed discovery, asset profiling, and vulnerability analysis.
- The **SAINT** vulnerability scanner offers the only integrated vulnerability assessment and penetration testing tools available.
- **Nmap** is a free, open source utility for network exploration or security auditing.
- Visit the **Metasploit site** for more information about the tool as well as tutorials and other resources.
- **WHAX** is the natural evolution of WHoppix—a live CD, stand-alone penetration testing toolkit from Softpedia.