Anatomy of a Web attack

Understanding the most common attack types

Sean-Philip Oriyano

February 03, 2009

More applications are being hosted on the Internet than ever before. Everything from databases to services is moving to a Web-based format. As a matter of course, this increased number of applications and services on the Internet has led to an ever-increasing number of attacks targeted at them. Learn how some of the more popular attacks work so that you can protect your organization.

Of the applications, Web sites, and services hosted on the Web, more than a fair share will experience some sort of mischief at the hands of a hacker intent on carrying out some sort of attack. To keep things short and sweet, I can easily say that although "The Truth may not be out there," people looking to deface, crack, exploit, break, steal, or otherwise mess with your site and application are.

Unfortunately, an increasingly sophisticated and hostile environment exists in today's Internet. In the case of those looking to harm your application, you have several things to consider. Basically, attackers have a lot of advantages that you as a defender don't. For example, attackers have a whole underground dedicated to sharing information as well as a (un)healthy desire to team up and create all sorts of havoc. Accentuating the threat is the fact that those wishing to "have a little fun" with your application have nearly limitless time, money, and resources. Couple these advantages with a questionable set of ethics, and the threat is even larger. Never underestimate or lose respect for those looking to do you harm: You do so at your own peril.

How likely are you to become a target? Well, the statistics can give one pause. Depending on the application's popularity, attacks can range from only a few an hour to several hundred or thousand in the same time period. In fact, the time to actually become a target of an attack or an attempted attack can be only moments after an application goes online and becomes available to the world. On average, a hosted application exposed to the Internet can experience over 400,000 different attack attempts (of varying degrees) over a one-week period.

Whether you're a definite target or not, it's safe to take the stance that you may become one. It is also wise to remember that if a service or application is placed on the Internet, it is at risk. What type of site or application you host can also greatly affect the type and frequency of attacks you will be exposed to. For example, if you host an online database application or e-commerce site, you
can expect to be squarely in the sites of an attacker at some point. If you happen to be hosting a blog, you're probably not a terribly large target (unless your blog is particularly popular).

This article looks at some of the more popular attacks and the types of attackers out there to see how they work so that you can understand how they might affect your organization.

**Attackers: Script kiddies and organized crime rings**

Most of the attacks that an organization will experience range from random, unstructured episodes to the well-organized and targeted variety—both of which tend to be automated. Automated attacks can vary in their relative complexity, with attacks being initiated against a target or opportunity directly, or (more likely) through several systems that may not even know they are being used as instruments in the attack. Estimates vary on how many systems may currently be compromised in such attacks, but it has been found that the systems used are present in all kinds of situations—from the small business to the large corporation.

Who typically is responsible for carrying out such attacks? In most cases, these automated attacks are launched by those with the lowest skill levels of the hacker community—those known as script kiddies. Script kiddies typically don't have the knowledge of those higher in the hacker community have, but that doesn't mean they can't be dangerous. When script kiddies launch an attack, they typically do so without realizing the results of their actions, such as potentially crashing systems or inadvertently performing a denial of service (DoS). These individuals fit the profile of a newbie who finds a new application, such as a scanner or password cracker, and runs it against large swathes of targets looking for an "interesting" result. It's not uncommon to see these kinds of attacks launched from networks hosted on college or university campuses toward targets in the "outside world."

The attacks that script kiddies launch look on the surface like those more organized groups or even the criminal element employ. In some cases, script kiddies are themselves pawns of organized crime or other organizations that might be looking to make financial gains or perform terrorist acts. They may even be used in fringe situations such as hacktivism.

### Definition

**Hacktivism** is using hacker techniques, such as Web defacement, denial of service attacks, and so on, against targets that are chosen for political purposes.

**Script kiddies, behind the scenes**

When considering script kiddies as threats, carefully consider what their goals may be. This information can yield valuable intelligence on how to protect yourself or your organization. In general, their goal is typically very simple, considering their technical experience: to seize control of a system or group of systems using the quickest and easiest techniques available. Additionally, script kiddies tend to select their targets based on ease of access and without regard to a system's relative importance or even whether that system is prone to crashing or other instability as a result of the attack.

Here's the typical attack process that a script kiddie uses:
• Construct a database of systems that are currently online and reachable. The attacker may obtain such a list by performing a simple ping sweep using any number of freeware tools, such as Angry IP scanner or Nmap.

• Scan each system discovered to uncover vulnerabilities, and move toward exploiting the vulnerability. The attacker can use tools such as Nessus or any number of utilities to perform such a scan.

• Gain control of the system. This step can be carried out in any of several ways, depending on the actual vulnerability.

• Hide the evidence. In this step, the script kiddie may employ a technique to cover up his or her tracks or the evidence of the penetration. However, in most cases, script kiddies neglect to perform this step.

• Modify the system. This step typically involves replacing core system files or changing the system configuration. In the case of script kiddies, it's typically done through some automated tool.

• Sit back and enjoy the chaos. With the system compromised, the attacker carries out whatever the next step may be—whether it's to gather data or compromise another system.

**Definition**

A rootkit is a program or set of programs designed to hide on a user's system and provide administrative control to an outside party. Rootkits will camouflage their components and destroy their traces by altering log files and erasing other tell-tale signs that an administrator might use to discover that a system had been compromised.

Keep in mind when reading this process that although the steps may seem complex, a script kiddie is more than likely using automated tools to perform the tasks. Many of the tools used to perform these tasks are readily available on the Internet and can be downloaded and run with little effort. What types of tools are available? Everything from those used to replace files to those that can enable remote control of a system, tracking of system activities, or even the very dangerous rootkit.

Also consider that in certain cases, script kiddies may post their results or actions on a newsgroup or blog, letting others know how and against whom they perpetrated their attack, thereby making you a bigger target. With a system compromised, an attacker may choose to pick any of a number of actions on the "menu," including attacking other systems or placing utilities on the system with the intent of waiting for valuable data to float by.

What kind of attack is this type of individual likely to carry out? In most cases, attackers at this level will attempt what is probably one of the simplest attacks: altering a Web site—classically known as a defacement. Their motivations for doing so may be legion. Individuals have been known to deface a Web site for reasons ranging from the pure heck of it all the way to gaining street "cred" or carrying out a grudge against a former employer. In a few rare cases, these individuals have even been known to perform attacks to extort a company or defraud it of money.

**The attacks**

When an attacker "paints" that target on your Web site or application's back, he or she typically has one of two goals in mind: either to adversely affect the ability of legitimate users to access the
site, or to lessen the reliability of a site. To accomplish these goals, attackers may enact an attack such as the traditional DoS or distributed denial of service (DDoS). An attacker may choose to adversely affect a site by defacing it, browsing around to locate valuable data, or dropping off an early Christmas present in the form of an infected file. Here are a few of the more "popular" attacks out there.

**Distributed denial of service**

In today's media, the DoS attack is frequently mentioned. But in most cases, what's really being referred to is a distributed DoS attack. The difference between these two is significant, both from a technical standpoint and from a threat profile standpoint. DDoS attacks are much more dangerous than their DoS brethren, as they involve a complex infrastructure in which an attacker compromises several systems that in turn are used to launch a coordinated attack against a victim. How many attacking systems are involved? Anywhere from hundreds to thousands to millions. Regardless of whether it's a single system or small number of systems, the result is catastrophic, and your application will probably be unavailable for a period of time.

Systems used in this type of attack can be compromised in any number of ways and infected with a worm or virus. In more complex situations, a rootkit may be dropped off and in turn used to launch an attack.

**Early bird gets the worm**

Another common way to compromise a Web site or application is through the use of worms. By definition, a worm is a self-contained software component that automatically performs probe actions designed to identify and exploit vulnerabilities on a targeted system. Additionally, a worm is designed to replicate as part of its life cycle, which in most cases means that it will scan, infect, and replicate over and over again, infecting more systems along the way.

In some cases, a worm may perform an additional function outside of simple replication: It may serve as a deployment mechanism for an enclosed payload. Various types of payload can be contained within a worm, but my example uses one of the more common and most dangerous payloads: the Internet Relay Chat (IRC) bot. When an IRC client is deployed onto a system, it can be used to issue commands to other parts of the IRC network, which may in turn be used to issue commands to other IRC clients that can attack as part of a DDoS scheme.

Keep in mind, however, that using a worm as a deployment mechanism for an advanced IRC-based network is not its only potential use. A worm can also deliver a payload in the form of a virus, a rootkit, or some other software device that modifies the system. The result, generically, is the same: A worm acts as an excellent attack vector for an attacker.

Another point to keep in mind with worms is that even a script kiddie can get access to any one of a sea of worms easily. If you're feeling ambitious, you can run a Google search for "worm source code." You'll find in excess of 1000 possibilities—in most cases, open source. Definitely a sobering thought to the defender of a network.
How do these attacks affect you?

After discovering the types of attacks available, consider how you can defend your organization. Although you can never fully eliminate the risks associated with putting a Web site or application online, you can be smart about it and address the weaknesses you may have. Let's take a look—albeit briefly—at the areas on which you should focus your defensive efforts.

The Web server

The defensive measures you can take on any particular Web server platform are numerous—much more than what I can cover here. But here are a few fundamentals.

The Web server should be one of the first places you focus your defensive efforts. When I refer to a Web server, I mean both the Web server application and the operating system on which it runs. Remember to look for misconfiguration, missing patches and hotfixes, and general weaknesses in the platform itself. While you're analyzing the server, also consider the hardware underneath and around it, as it forms part of the problem and the solution. Studies show that infrastructure such as routers, switches, and the like, is becoming a more tempting target. Having the hardware compromised can be just as devastating as the software.

Services and applications

Every system—whether it comes from a popular vendor or is open source—has services and applications installed and running that an attacker could exploit. Learn your platform's details and what to secure so that you harden your system against attacks.

Content

Ah yes, the high-profile part of the equation: what the attacker is generally looking for—that nice juicy content on your server. An attacker will go after content such as online databases with the intention of getting valuable information that can be used later—for example, credit card or customer information. Always learn your environment and how you can protect your content.

The pesky user problem

Now, the last thing worth mentioning is what's known as the carbon-based element—what I love to call the Layer 8 problem (you'll get it later)—the user. Never forget that the user and support staff working inside your organization are just as tempting a target as the vulnerabilities in the technology getting hacked through social engineering and, unfortunately, ignorance. To protect yourself and your organization from these threats, consider properly training your staff and users on proper security practices.

Conclusion

Today, more and more applications are being hosted on the Internet. As organizations have moved their applications to this environment, the threats have only increased. To avoid becoming a victim,
understand what you face, how attacks are carried out, and how you can employ proper defensive measures. A little research and understanding go a long way.

© Copyright IBM Corporation 2009
Trademarks
(www.ibm.com/developerworks/ibm/trademarks/)