

Internet Security Issues & Threats

“How Hackers Break In”

© 2003 Business Reform

“How Hackers Break In” by Russ McGuire, appeared in “BizNetDaily Commentary” for June 13, 2003.

BizNetDaily Editor's note: Russ McGuire is the online director of Business Reform Magazine. Each issue of Business Reform features practical advice on operating successfully in business while glorifying God.

One of my systems was hacked into a few weeks ago. Thankfully, I had a full backup and was able to restore the system without losing any data. But, it took at least a full day of my time to reload the operating system, apply all security patches, restore my data, and then reconnect to the Internet. I know my experience isn't unique, and I'm sure that many who are hacked into aren't as blessed as to be able to restore all of their data. In fact, more than 40,000 computer security incidents were reported to the Computer Emergency Response Team Co-ordination Center (CERT/CC) at Carnegie Mellon University in just the first 3 months of this year. Each incident can represent hundreds or even thousands of computers, and it is believed that the vast majority of security incidents are not reported.

Obviously, computer security is a big problem. I've never hacked into a system – in fact, I'm one of those people who can't even guess the passwords I've set for myself at different web sites and am constantly needing to request the password hint. So, if you're reading this column in hopes of "learning from an expert", you'll be sorely disappointed. However, I hope to provide some basic background on the methods that hackers use to gain unauthorized access to systems. Hopefully this will help you better understand the threats to any systems you may have connected to the Internet and understand your need for a robust firewall.

The vast majority of hacking attempts on the Internet today are done by fairly unsophisticated hackers often referred to as "script kiddies." These malicious individuals typically have no motivation other than to see how much damage they can do. They almost never go after a specific target, but rather try running their "scripts" against every single computer on the Internet (or at least the small subset of those they get around to before they get bored). Therefore, if your system has been hacked into, you can certainly take it personally since it's your life and perhaps your business that has been disrupted, but you shouldn't waste too much time trying to figure out what computer capable person who doesn't like you might be responsible – more than likely, you were merely a random victim.

Script kiddies take their name from the scripts that they typically run to break into systems. These scripts are typically written by a more sophisticated hacker and then posted on the Internet for anyone to download or e-mailed to friends. Think of it as the "Amway" of computer crime – a multi-level system where the top guy in the pyramid distributes his script to his hacker buddies who then pass it on to their hacker buddies, and so on. Each

level takes some sick pleasure from seeing how much damage was done by all the hackers downstream from them.

How can systems be hacked?

To understand how hackers break into systems, you need to understand how the Internet works. Given the space I'm allowed in this column, this will be a simplified version, but I hope it helps. Internet applications generally work using a client-server model. Some client software (e.g. your web browser) talks to some server software, usually on a different machine somewhere else on the Internet (e.g. Google's web server) to accomplish a task (e.g. find the Whit's End website your kids have been bugging you about).

There are lots and lots of these applications. You're probably most familiar with your web browser and your e-mail client, both of which follow this model. But even when using these applications, you're also using other client-server applications without even knowing it. For example, you may type into your web browser "www.whitsend.org", but what your web browser really needs is the IP address of that web server – something like "10.6.18.1". To get that address, your web browser automatically uses another application called the Domain Name Service (DNS) to send a request to a nearby Domain Name Server to find the IP address for "www.whitsend.org". You never see any of that happening, but it's happening.

Other applications you may have used, either consciously or not, include Telnet, File Transfer Protocol (FTP), Finger, WhoIs, News, Gopher... the list goes on. Bottom line, on any system connected to the Internet, some of these applications will likely be running at all times and they are designed to talk to other applications on other machines across the Internet. Often, when these applications are contacted by another machine on the Internet, they respond appropriately, sometimes writing information to the local hard drive. This is especially true for computers that are operating as servers. Applications like the FTP server or the DNS server or the e-mail server often need to have full system access to take appropriate action when they are contacted either by their corresponding clients or by other servers in the network.

This is the small crack that hackers need to break into a system. If the server software isn't carefully written, a skilled hacker might figure out a way to get the server to write something in a system file that gives a hacker full access to the entire system (for example, overwriting the system administrator's password, or creating a new user account). Or, maybe a hacker will figure out how to get the server application to crash in a way that allows the hacker to take over the application's session on the server, complete with full access to all files. This is not a good thing.

There's one more important aspect to how the Internet works that plays into defending against these attacks. The Internet is built upon three basic sets of protocols that are used for virtually all communications across the Internet. These are the Internet Protocol (IP), the Transmission Control Protocol (TCP), and the User Datagram Protocol (UDP). I won't bore you with details about these protocols, but in general, each application will only be

listening to a specific "port" through a specific protocol (e.g. UDP or TCP). A port is like a virtual channel that identifies the type of information that is included in an IP packet. The Internet community has defined specific ports to be dedicated to specific applications.

How do hacker scripts work?

Hackers go around looking for the kind of bugs I described above that can give them access into a system. The software bug will probably only exist in the specific application (e.g. the web server application) running on a specific operating system (e.g. Windows NT 4.0). Once they figure this out, it's typically referred to as an "exploit." Once, for example, Microsoft finds out about an exploit in the web server software for Windows NT 4.0, they will fix the bug and let everyone running that software know they need to download and install the patch on their system. Until Microsoft figures it out, all those systems are ripe targets for hackers. Even after Microsoft delivers the patch, all of the systems that have not yet been patched are likely targets of script kiddies.

Specifically, this is what a hacker script will do. It will pretend to be a certain type of application – for example, it will pretend to be a web browser. It will speak to a web server application using the defined port for web traffic (usually port 80). It will send the messages required to break open the system. Once the system is broken, it will record the information required for the hacker to come back later and log in to the hacked system and do as he pleases.

Why is it so hard to catch hackers?

One of the really nasty things about hackers is that, once they break into one system, they'll use it as a launching pad to start attacks against other computers. Therefore, if the system administrator of one system notices that he's under attack and can trace back the attack to another computer; chances are that computer was merely an earlier victim. Tracing the path all the way back to the original hacker is often impossible.

What can I do to prevent attack?

Nothing. If you have systems connected to the Internet, they will be attacked. If you're smart and blessed, those attacks will fail. I can't help you with the blessed part – that's between you and God, but there are some smart things you can do to minimize the chances you'll be successfully hacked:

1. **Don't run applications that you don't need.** If you're running an Internet server, but that server isn't used as a mail server, then don't run the mail server applications. Think of it as minimizing the number of doors that the hackers can use to break into your system.
2. **Buy and install a decent firewall.** These come in many different flavors and can either be really cheap (there's one built into the wireless access point I just bought for less than \$100) to really expensive (you could probably spend \$100,000 on one if you wanted to). The differences in price deal with the complexity of the type of

attack that can be repelled, the different types of attacks that can be survived (e.g. the Denial of Service attack doesn't try to break in to your systems at all, it just floods your network with traffic – this requires a different kind of solution than anything I've discussed so far), and performance (Google's firewall needs to handle a few more requests than mine without slowing people down...).

3. The firewall that you buy should minimally allow you to shut down specific ports that you don't need, and minimize the exposure of those ports that you leave open. For example, if you are running an e-mail server that sends and receives e-mail from the Internet, then you need to leave open the ports for SMTP (simple mail transport protocol). But if all of the people who actually get e-mail at this server are inside your network, then you can shut down the ports for the post office protocol (POP) used by e-mail clients to pick up and drop off e-mail at the server. That way, no one outside your network can use an exploit based on POP against your server. Similarly, if there's an application that only should be talking to one or two external servers (for example, DNS), then shut down those ports for all traffic except traffic originating from the one or two other IP addresses.
4. Turn on logging in your firewall and watch the logs. When you notice increased attempts to break through your firewall, that probably means that someone has recently introduced a new script that's customized for your type of system and you should be particularly aware of any strange behavior on your system.
5. Keep your system fully patched with the latest software updates from your system manufacturer and from the software developers of any server applications that you've installed.
6. Keep good backups. Since it may be days or weeks before you realize that you've been hacked, recognize that you might need to go back that far to restore your system to a safe state.

Unfortunately, even taking all of these precautions doesn't prevent you from being hacked. But it should greatly reduce the chances, and keeping good backups will greatly reduce your headaches and downtime – leaving more time to master the Train Game at Whit's End!

Russ McGuire is Online Director for Business Reform. Prior to joining Business Reform, Mr. McGuire spent over twenty years in technology industries, performing various roles from writing mission critical software for the nuclear power and defense industries to developing core business strategies in the telecom industry. Mr. McGuire is currently focused on helping businesspeople apply God's eternal truths to their real-world business challenges through Business Reform's online services. He can be reached at RMcGuire@businessreform.com.

Provided by: www.CommerceConnections.com ®