Know Your Enemy

The Tools and Methodologies of the Script Kiddie

Honeynet Project
http://project.honeynet.org
Last Modified: 21 July, 2000

My commander used to tell me that to secure yourself against the enemy, you have to first know who your enemy is. This military doctrine readily applies to the world of network security. Just like the military, you have resources that you are trying to protect. To help protect these resources, you need to know who your threat is and how they are going to attack. This article, the first of a series, does just that, it discusses the tools and methodology of one of the most common and universal threats, the *Script Kiddie*. If you or your organization has any resources connected to the Internet, this threat applies to you.

The Know Your Enemy series is dedicated to teaching the tools, tactics, and motives of the blackhat community. Know Your Enemy: II focuses on how you can detect these threats, identify what tools they are using and what vulnerabilities they are looking for. Know Your Enemy: III focuses on what happens once they gain root. Specifically, how they cover their tracks and what they do next. Know Your Enemy: Forensics covers how you can analyze such an attack. Know Your Enemy: Motives, uncovers the motives and psychology of some members of the black-hat community by capturing their communications amonst each other. Finally, Know Your Enemy: Worms at War covers how automated worms attack vulnerable Window systems.

Who is the Script Kiddie

The script kiddie is someone looking for the easy kill. They are not out for specific information or targeting a specific company. Their goal is to gain root the easiest way possible. They do this by focusing on a small number of exploits, and then searching the entire Internet for that exploit. Sooner or later they find someone vulnerable.

Some of them are advanced users who develop their own tools and leave behind sophisticated backdoors. Others have no idea what they are doing and only know how to type "go" at the command prompt. Regardless of their skill level, they all share a common strategy, randomly search for a specific weakness, then exploit that weakness.

The Threat

It is this random selection of targets that make the script kiddie such a dangerous threat. Sooner or later your systems and networks will be probed, you cannot hide from them. I know of admins who were amazed to have their systems scanned when they had been up for only two days, and no one knew about them. There is nothing amazing here. Most likely, their systems were scanned by a script kiddie who happened to be sweeping that network block.

If this was limited to several individual scans, statistics would be in your favor. With millions of systems on the Internet, odds are that no one would find you. However, this is not the case. Most of these tools are easy to use and widely distributed, anyone can use them. A rapidly growing number of people are obtaining these tools at an alarming rate. As the Internet knows no geographic bounds, this threat has quickly spread throughout the world. Suddenly, the law of numbers is turning against us. With so many users on the Internet using these tools, it is no longer a question of if, but when you will be probed.

This is an excellent example of why security through obscurity can fail you. You may believe that if no one knows about your systems, you are secure. Others believe that their systems are of no value, so

why would anyone probe them? It is these very systems that the script kiddies are searching for, the unprotected system that is easy to exploit, the easy kill.

The Methodology

The script kiddie methodology is a simple one. Scan the Internet for a specific weakness, when you find it, exploit it. Most of the tools they use are automated, requiring little interaction. You launch the tool, then come back several days later to get your results. No two tools are alike, just as no two exploits are alike. However, most of the tools use the same strategy. First, develop a database of IPs that can be scanned. Then, scan those IPs for a specific vulnerability.

For example, lets say a user had a tool that could exploit imap on Linux systems, such as imapd_exploit.c. First, they would develop a database of IP addresses that they could scan (i.e., systems that are up and reachable). Once this database of IP addresses is built, the user would want to determine which systems were running Linux. Many scanners today can easily determine this by sending bad packets to a system and seeing how they respond, such as Fyodor's nmap. Then, tools would be used to determine what Linux systems were running imap. All that is left now is to exploit those vulnerable systems.

You would think that all this scanning would be extremely noisy, attracting a great deal of attention. However, many people are not monitoring their systems, and do not realize they are being scanned. Also, many script kiddies quietly look for a single system they can exploit. Once they have exploited a system, they now use this system as a launching pad. They can boldly scan the entire Internet without fear of retribution. If their scans are detected, the system admin and not the black-hat will be held liable.

Also, these scan results are often archived or shared among other users, then used at a later date. For example, a user develops a database of what ports are open on reachable Linux systems. The user built this database to exploit the current imap vulnerability. However, lets say that a month from now a new Linux exploit is identified on a different port. Instead of having to build a new database (which is the most time consuming part), the user can quickly review his archived database and compromise the vulnerable systems. As an alternative, script kiddies share or even buy databases of vulnerable systems from each other. You can see examples of this in Know Your Enemy: Motives. The script kiddie can then exploit your system without even scanning it. Just because your systems have not been scanned recently does not mean you are secure.

The more sophisticated black-hats implement trojans and backdoors once they compromise a system. Backdoors allow easy and unnoticed access to the system whenever the user wants. The trojans make the intruder undetectable. He would not show up in any of the logs, systems processes, or file structure. He builds a comfortable and safe home where he can blatantly scan the Internet. For more information on this, check out Know Your Enemy: III.

These attacks are not limited to a certain time of the day. Many admins search their log entries for probes that happen late at night, believing this is when black-hats attack. Script kiddies attack at any time. As they are scanning 24hrs a day, you have no idea when the probe will happen. Also, these attacks are launched throughout the world. Just as the Internet knows no geographical bounds, it knows no time zones. It may be midnight where the black-hat is, but it is 1pm for you.

This methodology of scanning for vulnerable systems can be used for a variety of purposes. Recently, new Denial of Service attacks have been reported, specifically DDoS (Distributed Denial of Service attacks). These attacks are based on a single user controlling hundreds, if not thousands of compromised systems throughout the world. These compromised systems are then remotely coordinated to execute Denial of Service attacks against a victim or victims. Since multiple compromised systems are used, it is extremely difficult to defend against and identify the source of the attack. To gain control of so many systems, script kiddie tactics are often employed. Vulnerable systems are randomly identified and then compromised to be used as DDoS launching pads. The more systems compromised, the more powerful the DDoS attack. One example of such an attack is 'stacheldraht',. To learn more about Distributed Denial of Service attacks and how to protect yourself, check out Paul Ferguson's site Denialinfo

The Tools

The tools used are extremely simple in use. Most are limited to a single purpose with few options. First come the tools used to build an IP database. These tools are truly random, as they indiscriminently scan the Internet. For example, one tool has a single option, A, B, or C. The letter you select determines the size of the network to be scanned. The tool then randomly selects which IP network to scan. Another tool uses a domain name (z0ne is an excellent example of this). The tools builds an IP database by conducting zone transfers of the domain name and all sub-domains. User's have built databases with over 2 million IPs by scanning the entire .com or .edu domain. Once discovered, the IPs are then scanned by tools to determine vulnerabilities, such as the version of named, operating system, or services running on the system. Once the vulnerable systems have been identified, the black-hat strikes. For a better understanding of how these tools are used, check out Know Your Enemy: Forensics.

How to Protect Against This Threat

There are steps you can take to protect yourself against this threat. First, the script kiddie is going for the easy kill, they are looking for common exploits. Make sure your systems and networks are not vulnerable to these exploits. Both www.cert.org and www.ciac.org are excellent sources on what a common exploit is. Also, the listserv bugtraq (archived at securityfocus.com) is one of the best sources of information. Another way to protect yourself is run only the services you need. If you do not need a service, turn it off. If you do need a service, make sure it is the latest version. For examples on how to do this, check out Armoring Solaris, Armoring Linux or Armoring NT.

As you learned from the tools section, DNS servers are often used to develop a database of systems that can be probed. Limit the systems that can conduct zone transfers from your Name Servers. Log any unauthorized zone transfers and follow up on them. We highly recommend upgrading to the latest version of BIND (software used for Domain Name Service), which you can find at www.isc.org/bind.html. Another option is to use djbdns as a replacement for BIND. Last, watch for your systems being probed. Once identified, you can track these probes and gain a better understanding of the threats to your network and react to these threats.

Conclusion

The script kiddie poses a threat to all systems. They show no bias and scan all systems, regardless of location and value. Sooner or later, your system will be probed. By understanding their motives and methods, you can better protect your systems against this threat.