

▶ Internet Penetrations Thinking Like an Attacker

A Perspective on Practical Security 2005
by Brad C. Johnson

© Copyright 2005 SystemExperts Corporation. All rights reserved.

► Internet Penetrations Thinking Like an Attacker

Abstract

This article describes the process of trying to penetrate your Internet based resources from the viewpoint of an attacker. This paper was prepared at the request of the editors of Security Enterprise Magazine for publication in April 2005.

Within the complex space of the IT environment, there are some common reasons and resources that help increase the success rate of an attacker such as intrusion detection, Web applications, 802.11, modems, and that there is easy to find and compelling data that can be used in an attack.

The attacker methodology is a straightforward four part process that includes reconnaissance (profiling), cataloging & prioritizing of the profiled data, attack research, and testing & validation of the attack scenarios.

The most important attributes in an Internet based attack are *diligence* and *vigilance*. Despite all the things working in the attacker's favor, there are some simple and effective steps you can take to make your security stance better.

Inside

- Understanding why Internet based attacks are so successful
- Looking at what the attacker knows about your environment
- A review of the typical attacker methodology
- An example attack
- Recommendations on how to be better prepared for Internet attacks

Introduction

This article describes the process of trying to penetrate your Internet based resources from the viewpoint of an attacker. Let's start with a couple of important definitions.

What does it mean to "penetrate" your Internet resources? It means that somebody – an attacker, a hacker, a determined intruder, an accidental intruder, a disgruntled employee or family member – gets access to either data or systems that were meant to be either private or restricted. You do not have to "break in" to a system to be successful in an attack.

For example, an attacker might be able to view critical network management data by using the SNMP protocol from a remote location on the Internet. An attacker might be able to get packets to go to the wrong system by using a DNS poisoning attack. An attacker might be able to get access to local files on a system in the internal network by using SQL injection techniques through your Web application.

There are an almost endless number of opportunities that do not require somebody obtaining interactive access on a machine to get access to data or systems that were meant to be private or restricted.

How does an attack really work? If a tool or technique gives access to the private/restricted data/systems we have just been talking about, it is a successful attack. It is just as simple as that. Sometimes the energy and expertise needed to achieve success is embarrassingly simple (e.g., download a tool and run it) and sometimes it is excruciatingly difficult. Almost all attacks are based on one simple premise.

The more an attacker knows about what is running on a site, the more likely it is the attacker can find or develop a tool or technique that will be successful. Understanding the details of what is on your network is called developing a profile.¹ If the attacker can figure out exactly what kinds of resources you have, exactly what systems and services are running on those systems, and how those devices are configured, the chance of being able to exploit some problem on it is very high. The following section explains why attacks are more successful than we would like.

1. An article at TeCrime International points out profiling as one of the critical steps in attacking a system (<http://www.tecrime.com/llartH09.htm>).

► Internet Penetrations Thinking Like an Attacker

Why attacking Internet Resources is so Successful and Easy

The simple reason that most Internet attacks are both easy² and successful is that it is difficult to integrate the various technologies that you need to run a normal production environment. In a word, today's security infrastructure is a *hodgepodge* of components. Some are mature, some are brand new, some are open source, some are proprietary, and they are inconsistent in how they deal with with the key attributes of security including authentication, authorization, and auditing.

Integrating these technologies from different companies is incredibly hard: especially when they have been built with different views of how security should work. Every single component in your network has its own set of potential vulnerabilities. The integration points themselves are also a source of even more problems.

To an attacker, all of those technologies and integration points are opportunities: opportunities for problems that can be exploited. These include development problems (e.g., buffer overflow attacks, command injection), configuration problems (e.g., using default settings, not changing well known credentials), or design and deployment problems (e.g., difficulty in integrating 3rd party intrusion detection with your Microsoft Services with your home-grown Web application).

What the Hacker Knows about your Environment

Within the complex space of the IT environment, there are some common reasons and resources that help increase the success rate of an attacker.

Inadequate Intrusion Detection

There are a wealth of tools, techniques, experts, courses, and books that are focused on intrusion

detection. Yet despite all of these resources, most Internet sites have either no intrusion detection infrastructure or if they do have something in place, it is prepared to notice only generic intrusion attempts.

The real problem is that attackers know this! There are a number of reasons for this dichotomy; here are a few of them.

- Fundamentally, intrusion detection is not as easy as you would hope. It requires a significant amount of configuration and testing time to deploy software that actually works. In addition, most intrusion detection software is not built to help with some of the most common types of actual intrusion areas such as modems, Web applications, and wireless components.
- The only way to develop a detailed understanding of what is "normal" in your environment is through an iterative process of capturing and analyzing production network traffic. This is one of the important reasons why most intrusions are not detected even if you are watching when it happens! People cannot tell the difference between "normal" traffic and an intrusion.
- There are dozens of incredibly easy to use tools that will look for or attempt to exploit hundreds of vulnerabilities. Five years ago, if the attackers were not actual experts in writing network programs, in understanding exactly how protocols worked, and also adept at low level operating system details, they would be hard pressed to be successful. In today's world, there are so many "Cut/Paste" utilities or instructions available, an attacker does not have to be an expert to be successful.

Web Applications

Problems in Web applications are the fastest growing exploit area.³ The Open Web Application

2. The Director of the CERT Center, before the House Committee on Government Reform in late 2002, points out that Internet attacks are inherently easy because many of the fundamental protocols were developed with an assumption of trust (http://www.cert.org/congressional_testimony/pethia-11-02/Pethia_testimony_11-19-02.html).

3. An article in CSO Online helps to articulate why standard security measures such as firewalls, host, and network services are not enough to protect you from Web application specific problems (<http://www.csoonline.com/read/050104/application.html>).

► Internet Penetrations Thinking Like an Attacker

Security Project (OWASP) has identified 10 common problems with Web applications that every company should be aware of. Unfortunately, most Web applications are not tested for these types of exploits. The reality is, however, when that application is deployed on the Internet, it must protect itself from these types of problems (that have nothing to do with the normal functional behavior of the application) or it may be the vehicle that an attacker uses to get access to important resources.

Attackers know that here in the early 21st century, one of the easiest ways to access data, that was intended to be private, is directly through a company's Web application using standard, off-the-shelf technology like a browser. Using a browser, the attacker can manipulate the pages (e.g., change the contents of state information in cookies or environment variables, insert commands or special characters in forms, or change the HTML page to have different information than was originally sent) that are returned to their system (i.e., the Web pages themselves) and send it back to the server and see how it responds.

802.11 Wireless

Wireless technology is one of the fastest changing parts of many organizations' network infrastructure. Most organizations deploy 802.11 wireless components because they are inexpensive to buy, easy to deploy, do not require sophisticated knowledge to install, and they allow you to extend your network without physical changes. Unfortunately, because wireless components are often deployed without either the advice or assistance of IT or security professionals, they are often used in their default configuration.

As almost every hacker knows, the out of the box condition is usually its least secure configuration.⁴ Additionally, the default parameters for these devices are often well known such as the default Linksys SSID for some 802.11b systems is "linksys"

and for 802.11g is "linksys-g" while the default SSID for some D-Link products is "wlan". There are a number of repositories of this information including the default SSID, channel number, WEP key(s), IP address, and the administrative password (e.g., http://mediawhore.wi2600.org/nf0/wireless/ssid_defaults/ssid_defaults-1.0.5.txt).

What may be even more important than the specific setup, is that without a serious effort to understand where the radio frequencies are actually going, the network is probably extending beyond the physical boundaries of what was intended: such as to the floor above or below you, to the building next to you, or to public places like the streets and roads around your building.

Modems

Modems are potentially the single easiest way to access private, internal information because they typically bypass every implemented security mechanism and yet exist on critical systems such as routers, firewalls, printers, and desktop systems. Your own administrators and vendor support staff often use them to remotely manage or monitor important services. Modem-based services are frequently not using any type of encryption, they often do not require any authentication other than dialing the number, and the actions are almost never logged.

In the world of hacking, this is one of the oldest types of attacks and one of the first exploits to be automated with what is referred to as War Dialers. These programs run unattended and dial phone numbers and look for recognized devices to attach to. When the program is finished, all the attacker has to do is look at the resulting list of numbers that had connection oriented modems (e.g., a printer, a router, an electrical system), dial that number again, and attempt to login. For some services, no username or password is required or the default configuration uses well known passwords.

4. A team from Federal Computer Week performed a wireless security survey in the Washington, D.C. area in late 2004 demonstrating a number of the problems with the default configuration of wireless services (http://www.usatoday.com/tech/news/computersecurity/infotheft/2004-11-09-fed-weakest-link_x.htm).

► Internet Penetrations Thinking Like an Attacker

Lots of Useful and Detailed Research Data and Resources

Probably the most underestimated fact about attacks is that there is a wealth of information available to anybody willing to invest a little time browsing the Internet. The information is readily available, compelling, and often provides incredibly detailed information that is useful in an attack.

The good news for both you and the attacker is that there are a number of organizations that are dedicated to making it easy to find out what exploits or vulnerabilities exist for specific devices, operating systems, applications, programs, languages, and any other network based resource.

Following are just a few examples of places anybody can go to get detailed information that can be used to construct, research, or think about a specific attack.

- General Internet security
www.cert.org,
<http://cve.mitre.org>,
www.osvdb.org
- Security archives
www.packetstormsecurity.org,
<http://xforce.iss.net>,
www.securityfocus.com,
<http://archives.neohapsis.com>
- General news
www.google.com
- Hackerz
www.defcon.org,
www.antonline.com,
<http://www.2600.com>
- Vendors
Microsoft, Sun, HP, IBM, Red Hat
- Operating System (OS) or Application Specific
www.ntsecurity.net,
www.ntbugtraq.com,
www.isc.org/bind.html,
www.dns.net/dnsrd

Attacker Methodology

The attacker methodology is a straightforward process. You start with a general hypothesis (Can I get access to data or systems that were intended to be private or restricted?), perform general research to learn about the target environment (i.e., develop a profile of exactly what components are used at the target site), use that research to refine the scope of what items are worth pursuing in more detail, and then iterate through the process of attempting various exploits and conducting finer grained research.

For an attacker, the four parts of that process are as follows:

1. Reconnaissance
 - Send packets to the target systems and learn how they are setup and what they are running
2. Catalogue & Prioritize
 - Take the reconnaissance data and determine what is worth researching in more depth
3. Research
 - Review available documentation, reports, release notes, configuration descriptions, specifications and do online research on who else has dealt with the specific component including known exploits or vulnerabilities
4. Test & Validate
 - Use the data, techniques, and tools discovered during the research and try to an actual attack or to learn more about the profile of the site

One of the interesting observations about the above methodology is that only the first and fourth steps require sending packets over the network to the target site. The second and third steps are “offline” in the sense that the work is done mostly over the Internet (e.g., Google searches and follow-up), but does not include sending packets to the target site.

What is especially interesting is that experience shows that most of the work in an attack is indeed in these second and third steps. Performing an attack does not require sending a lot of data to the target systems. See the following Figure for a view of this attacker methodology.

Internet Penetrations Thinking Like an Attacker

Attacker Iterative 4-Step Methodology:

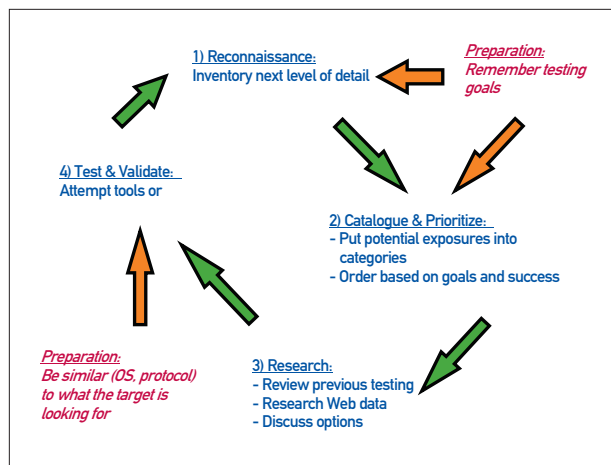


Figure 1: The Attacker Methodology Process

Attack Example

So far, we have been talking about how straightforward a lot of the activities an attacker would use are. Just like a picture can be worth a thousand words, the following example should highlight the main points in this article.

Cisco is one of the most widely used vendors for networking technology. In particular, it is the world leader in network router technology so it would not be surprising for an attacker to think about trying to use a Cisco router as an attack point. How would an attacker do that?

Most attackers use public-domain and third-party tools to help with the profiling part (Step 1 of the Attacker Methodology: Reconnaissance) of the attack process. There are literally hundreds of profiling/scanning tools that an attacker can use, and many of them are free. One popular open-source profiling tool is called Nessus (<http://www.nessus.org>).

Nessus will scan a set of systems and attempt to find and report on over 6,000 tests (yes that is six thousand) for different vulnerabilities. Assuming you have downloaded the Nessus tool, let's see how difficult it is to find a problem and develop an

attack against a Cisco router. Okay, I will tell you ahead of time, it is not difficult.

Step 1: Run Nessus which discovers at least one Cisco router

Step 2: Use your browser and the Google search engine (www.google.com) and put in the words "Cisco Hacking" and on the first page of links returned will be a reference to an article published by Computer World⁵

Step 3: Read the first page of the article in Computer World and see a reference to something called the "Cisco Global Exploiter", a tool that can find specific exploits on various Cisco routers

Step 4: Use your browser again and put in the words "Cisco Global Exploiter" and on the first page of links returned will be a variety of references to sites that have the actual code for this program⁶

Step 5: Make a copy of the Perl script program and run it on some host on your network (see the following Figure for a small portion of the code showing the various exploits it can attempt)

Cisco router exploit:

```

"Vulnerabilities list :";
"[1] - Cisco 677/678 Telnet Buffer Overflow Vulnerability";
"[2] - Cisco IOS Router Denial of Service Vulnerability";
"[3] - Cisco IOS HTTP Auth Vulnerability";
"[4] - Cisco IOS HTTP Configuration Arbitrary Administrative Access Vulnerability";
"[5] - Cisco Catalyst SSH Protocol Mismatch Denial of Service Vulnerability";
"[6] - Cisco 675 Web Administration Denial of Service Vulnerability";
"[7] - Cisco Catalyst 3500 XL Remote Arbitrary Command Vulnerability";
"[8] - Cisco IOS Software HTTP Request Denial of Service Vulnerability";
"[9] - Cisco 514 UDP Flood Denial of Service Vulnerability";
  
```

Figure 2: Code sample from "Cisco Global Exploiter"

That attack took five steps.

5. <http://www.computerworld.com/securitytopics/security/story/0,10801,91748,00.html>

6. <http://www.hackwire.com/comments.php?id=62&catid=2&highlight=>

► Internet Penetrations Thinking Like an Attacker

The attacker did not have to be a Perl coding expert, they did not have to be a Cisco router expert, and they did not have to understand anything about how the Nessus tool works but within a few hours they might have been able to execute arbitrary commands on your Cisco router.

If you also looked at a few of the other links returned by Google in this simple example, you would have found other interesting and fruitful references like an article posted on Security Focus⁷ which is a two-part article focused on “Identifying and Exploiting Vulnerabilities and Poor Configurations in Cisco routers” or that a hacking conference named Black Hat is offering a course entitled “Hacking Cisco Networks”⁸ this year at its European conference.

Final Word

What does it take to be a successful attacker? The most important attributes are *diligence* and *vigilance*. There are also a number of factors that are working in the attacker’s favor.

- It is hard to get your intrusion detection resources working well enough to notice anything more than the most simplistic intrusion attempts
- The fastest growing area for attacks is also the fastest growing area for most businesses: using Web applications on the Internet
- Most organizations are using technologies that are inherently easy to attack such as wireless setups and modems
- Integrating a diverse set of technologies to create a well secured environment is extremely difficult and is likely to create even more opportunities for problems that can be exploited
- It is easy to do research on potential attack points

and finding out what a site has (profiling) can be done with free and easy-to-use tools

- Most successful attacks do not require sending a lot of data to the target systems (making the detection of the attack even harder)

To combat the attackers, it would be wise to adhere to Occam’s Razor, which loosely means that all things being equal, the simplest approach is likely to be the best approach. Here are 10 simple, yet extremely effective, pieces of advice to help make your network more secure.

1. Change the default passwords for all wireless, SNMP, Web server, and Web services you have
2. Make a copy (clone) of critical systems, services, or configurations as soon as they are setup correctly
3. Make a list of the 10 most likely reasons you would lose your job over an attack and make sure you either have technology to prevent it or processes to detect it (even if it is after-the-fact)
4. Reduce the complexity of your environment by removing extraneous systems, services, and executables
5. Reduce the variety of services you offer (e.g., run 2 different Web servers instead of 4, support 3 Windows OS types instead of 6)
6. Regularly run vulnerability assessment tools against all of your systems
7. Run a war dialer against your resources that can be reached via telephone
8. Survey your wireless environment to see how far it actually reaches
9. Test your networked based hosts for the SANS Top 10 problems⁹
10. Test your Web applications for the OWASP Top 10 problems¹⁰

7. <http://www.securityfocus.com/infocus/1734>

8. <http://www.blackhat.com/html/bh-europe-05/train-bh-eu-05-sd.html>

9. <http://www.sans.org/top20>

10. <http://www.owasp.org/documentation/topten.html>