Stealing The Network: How To Own The Box

"Stealing the Network: How to Own the Box is a unique book in the fiction department. It combines stories that are false, with technology that is real. While none of the stories have happened, there is no reason why they could not. You could argue it provides a road map for criminal hackers, but I say it does something else; it provides a glimpse into the creative minds of some of today's best hackers, and even the best hackers will tell you that the game is a mental one."

- from the foreword by Jeff Moss, President & CEO, BlackHat, Inc.

Chapter 5 is excerpted from the book titled "Stealing The Network: How To Own The Box" By Ryan Russell, Tim Mullen (Thor), FX, Dan “Effugas” Kaminsky, Joe Grand, Ken Pfeil, Ido Durbrawsky, Mark Burnett, Paul Craig, published by Syngress. ISBN: 1931836876; Published: April, 2003

The Thief No One Saw

By Paul Craig

This is my story. My name is Dex. I’m a 22-year-old systems administrator. I live in an upper-class apartment in New York’s CBD. My apartment is lined with computers, coffee cups, and cables. I work eight hours a day for a small online e-commerce site, mostly managing servers and security.

In my free time, I run my own contract development company, writing mostly C/C++. I also moonlight as a “Rent a Thief ”for a black market media “distribution” company based out of Taiwan. On demand, I hack into companies and steal whatever is required. Usually, it’s a new, highly anticipated game or a large, expensive CAD (computer-aided design) software package. Once, I was even asked to steal software used to design a nuclear power plant. I don’t ask questions. This thievery doesn’t stop at software, though. There is big money in commercial plans, financial data, and customer contact lists, as well…

I do this because I enjoy the rush and the feeling of outsmarting someone else. I never tell anyone else about a hack, and to date, only a few companies I’ve hit even suspected that they had been hacked. I am not a part of the typical “hacker” community, and I always work alone.

The Tip-off

My eyes slowly open to the shrill sound of my phone and the blinking LED

in my dimly lit room. I answer the phone.

“Hmm … Hello?”

“Yo, Dex, it’s Silver Surfer. Look, I got a title I need you to get for me. You cool for a bit of work?”

Silver Surfer and I go way back. He was the first person to get me into hacking for profit. I’ve been working with him for almost two years. Although I trust him, we don’t know each other’s real names. My mind slowly engages. I was up till 5:00 A.M., and it’s only 10:00 A.M. now. I still feel a little mushy.

“Sure, but what’s the target? And when is it due out?”

“Digital Designer v3 by Denizeit. It was announced being final today and shipping by the end of the week, Mr. Chou asked for this title personally. It’s good money if you can get it to us before it’s in the stores. There’s been a fair bit of demand for it on the street already.”

“Okay, I’ll see what I can do once I get some damn coffee.”

“Thanks dude. I owe you.” There’s a click as he hangs up.

I know of Denizeit very well. In fact, I’ve wanted to get a hold of some of their software for quite some time. They make high-end, commercial, 3D design/postproduction software used in many large-scale animated movies and games. Their stuff is like digital gold. The thrill of stealing the software that was used to make the bullets appear to stop in The Matrix will be more than worth the effort and risk involved. This will be a very nice trophy to add to my collection.

Once my client (Mr. Chou) gets his hands on the software, he will be printing a few thousand CDs of it and selling them on the street before Denizeit is able to ship the product to stores. This must happen before it’s shipped to stores, so he can be the only person in the world selling it. Mr. Chou doesn’t care about what the product looks like. If it doesn’t have the correct CD labels, manuals, or boxes, that’s just fine. He just wants the product on CD/DVD.

My fee is 10 percent of the amount sold in the first two months. A title like this might sell 2,000 to 5,000 copies easily on the street. The black market price sits at about $10 to $20 (US) a copy, which is very reasonable, given the retail price for a legal copy is $4,000. So, I should make around $5,000 (tax free).

A company like Denizeit will by no means be easy to break into, and I will not be the first hacker to have tried. My attack has to be thought out, logical, and executed very methodically. I quickly devise a mental plan/checklist of the approach I’ll take:

-

Gather as much information as possible about not only Denizeit’s network and hosts, but also company structure, organizational charts, phone numbers, on-call rosters, and especially any laid-out “best” practices for IT security response.

- Obtain as much possible information about the software—what developers are working on it, where they are located, what hours they work, whether they work from home, which operating system (OS) they use. Do they drink their coffee with cream or milk?

- Gather internal news releases and obtain the final build number of Digital Designer.

- Plan my attack—what hosts I’ll use, when I’ll use them, and who I’ll log in as. Prepare everything and work to a very strict time limit. Although this is hardly Mission Impossible, the jail term associated with it is very real.

- Obtain all software and ship CDs. I have just under four days to get the CDs out. I should really have them shipped by tomorrow afternoon at the latest.

Studying the Prey

At this point, most hackers who wanted to break into a host would simply fire up a suite of penetration-testing tools and begin to scan for known vulnerabilities. Programs like nmap, Whisker, retina, and the like will quickly find an exploitable application or insecure port.

However, since I don’t know if this company has a firewall or IDS yet, the last thing I want is for the security admin to be woken up at 5:00 a.m. because he gets an SMS alert saying that someone is trying to break into his servers. Chances are, if he doesn’t suspect an attack, he won’t be looking for me and probably won’t see me snooping around. Any premature tip-off may also spark a quick server security check. I want this network to feel safe and cozy to the folks running it, and if I do my job right, they’ll never even know I was there.

The first thing I do is look at the company’s Web site. I read it, studying its every minor detail and learning as much as possible from it. A Web site is very much the clothes of a company. You can tell a lot by looking at someone’s clothes: what kind of neighborhood they most likely live in, how much money they make, how much they care about appearances, and whether they want everything to be perfect.

www.denizeit.com is a well-designed site, quick loading, and easy to navigate. This isn’t a small outfit, and their site looks very professionally done. It’s also massive; it must have around 100 ASP pages full of content, support, knowledge bases, press releases, and product information. One interesting thing is that everything appears to be on www.denizeit.com, so it looks like there is just one big, powerful server. I see no signs of separate server names, such as support.denizeit.com or news.denizeit.com. Maybe they have bought some hosting space somewhere, or perhaps this is a just a single, large server or a cluster of servers behind a load balancer of some kind.

An interesting question to ask is, “Is this site developed in-house or contracted out to an external development company?” If the content of the site is going to be changing regularly, or there is a large amount of content to manage, it probably will be developed in-house. Managers hate having to pay Web design consultants every time they want a small change made; it’s a lot easier to have a few Web developers on staff.

My guess is that Denizeit has one or two full-time Web developers, since there is a fair bit of dynamic code on the site, such as searching support, e-mail forms, and so on, and these are also all written in ASP. I am also sure that, being a graphic design company, there would be no shortage of graphic designers on staff. A site like this would require at least one full-time graphic designer.

This also leads me to think about their Web server architecture. A large company with a large Web site like this would be very worried about risk and would probably have a development site somewhere—at a guess, I would say something named staging.denizeit.com or development.denizeit.com. Chances are this should be located internally behind a firewall and accessible only by the support staff. However, external live development sites are very common these days.

The reason I think about a development site is that I have yet to see a development server that has the same level of security as a live Web server. People simply forget about the staging server when it comes to upgrades and patches, and

log files may be discarded and unchecked for security breaches.

Now, to dig a little further, I do a WHOIS request on www.denizeit.com. All I want to gain here is the name of the system administrator or person who is responsible for setting DNS names up. It should also list his phone number. This information isn’t really a big deal to get; usually, a quick search of a site will turn it up, but knowing something as simple as a name can often help you become familiar with an alien network.

WHOIS Record

Domain name: denizeit.com

Name servers:

ns.denizeit.com

ns2.denizeit.com

Created: 10/02/2002 14:46:23

Expires: 10/02/2004 14:46:23

Registrant Contact:

Andrew Jacob

ajacobdDenizeit.com

New York, NY 89134

US

702 804 1955

702 804 1956

Administrative Contact:

Andrew Jacob

ajacob@denizeit.com

New York, NY 89134

US

702 804 1955

702 804 1956

The WHOIS record shows Andrew Jacob, American-based, as the sysadmin. I guess if all else fails, I can call him and ask for his root password, I laugh to myself.

I look out my window, noticing that the sun is now shining directly into my eyes. Damn! I hate the light. It really burns when you prefer the darkness. I shut my blinds and turn on my dim, red light bulbs. God bless the person who invited red light bulbs. They have saved me many a headache.

The DNS Giveaway

My first task now is to have a general look at their network from a very high-level DNS point of view. Basically, I want to find out what kind of DNS entries they have set up. A typical network might have something like this:

www.example.com

mail.example.com

ns.example.com

ftp.example.com

This is a very easy way to get a nice clean map of a company’s network. The average company will name their gateway gateway, their FTP site ftp and their development server dev. It’s only logical that they do so, but it also allows me to focus an attack quickly, without the need for port-scanning or any intrusive method to determine a server’s primary task.

I can also glean a fair bit of information about network architecture by simply looking around on a site. If I had seen that the WHOIS record for the DNS name was registered to a contact in France and the Web server’s IP address was

also located in France, but their support site was located in Germany, I could assume that the company had branches in both Germany and France. It’s possible they outsource their support to a different company or branch, in which case, they’re likely to have some smaller networks in each location. Chances are these networks need a way to talk to each other. So they probably run a VPN of some kind or use a lot of e-mail communication.

So what’s an easy way to obtain a DNS “map” of a hostname/network? I could request a zone transfer for the domain of www.denizeit.com from their DNS server (ns.denizeit.com). If their DNS server allowed me to do this, I would be able to find every host on their network in one hit. However, a lot of common IDSs these days detect zone transfers and report them as being suspicious.

The other way would be to simply attempt to resolve a list of common DNS names using a tool I wrote called DNSMAP. With this little program, I’m able to do a reverse DNS lookup for a few hundred DNS names in a short amount of time; for example, trying to resolve mail.denizeit.com to an IP address, then www2.denizeit.com, smtp.denizeit.com, and so forth. These will look like common DNS lookups, unsuspicious to the untrained eye. It will also allow me to find other possible IP subnets they have lurking around.

I decide that since I’m still unsure of what security architecture Denizeit has, I’ll use DNSMAP to attempt to passively resolve their network. Although I may be what some people think of as a renegade/carefree hacker, I’m actually very scared of going to jail. Plus, I take a certain pride in not being seen.

Output of DNSMAP on denizeit.com

root@lsd root]# dnsmap denizeit.com

DNS Network Mapper v1.1 © Dex

Searching subhosts on domain denizeit.com

mail.denizeit.com

IP Address #1:61.101.28.34

www.denizeit.com

IP Address #1:209.151.252.38

IP Address #2:209.151.252.73

ftp.denizeit.com

IP Address #1:209.151.252.38

IP Address #2:209.151.252.73

ns.denizeit.com

IP Address #1:209.151.252.16

ns2.denizeit.com

IP Address #1:209.151.252.16

firewall.denizeit.com

IP Address #1:61.101.28.41

vpn.denizeit.com

IP Address #1:61.101.28.34

[root@localhost root]#

This produces a virtual gold mine of information for me! I can see that their WWW and FTP servers have two IP addresses assigned to them. This could be a DNS round-robin to provide some load balancing, or maybe just a backup IP address for fault tolerance. At first glance, I also see that they have two different IP classes: 209.151.252.xx and 61.101.28.xx.The most likely reason for this is that their WWW and FTP servers are hosted at a large colocation point, one with some serious bandwidth and network reliability (which would explain the dual IP addresses on www.denizeit.com). The 61.101.28. class is probably a leased line to their main office.

It would make sense for them to have their VPN, firewall, and mail server as close as possible to the core user network. A quick check of what OS the Web server is running will give me a little more information on what their OS of choice is. For this, I telnet to port 80 and issue a manual HTTP GET that would look like someone has mistyped a URL (in this case, http://www.denizeit.com/index.htmx). This will cause the server to return a 404, and in the header of the HTML response, I should get the server response. There are a lot of ways to do this, but I find this to be the most unobvious way. I really like to be sleek in the way I work.

WebServer Check

GET /index.htmx HTTP/1.0

HTTP/1.1 404 Object Not Found

Server: Microsoft-IIS/5.0

Date: Sun, 23 Mar 2003 11:19:33 GMT

Content-Type: text/html

I see the server is listed as IIS5.That’s probably a Windows 2000 Server. Although it’s possible to change or fake your server’s return headers, most people don’t do it. So, it’s a safe guess that this is a Windows box, especially since they have so many ASP pages.

A quick read-through of their Web site shows that they develop their software for only Microsoft Windows 2000; there’s no Linux or UNIX support of any kind. I would guess that almost all the machines on this network are Windows-based. There might be one or two Linux or UNIX machines—most likely the name server and perhaps the odd client PC running Linux (for the daring, challenging few). I could be totally wrong about this, but seeing the amount of work that was put into their Web site (all written in ASP), and given the fact that this Web site is their main client-facing element, chances are they would use something that they really liked and trusted. If the company was not 100 percent sure of Windows, they would not use it for a Web server. If you were comfortable with Windows for such an important role in your network, chances are you would use it for other tasks as well. This allows me to target my attack more precisely. Attacking a UNIX server is a very different task than attacking a Windows server.

It’s lunchtime now, and my mind is becoming a little buzzed with the anticipation of this hack. I can feel it will be a good one. However, after noticing firewall.denizeit.com, I’ll need to be careful. Although I have not been caught yet, there’s always a first time. But it’s nice to know that Denizeit decided to call the firewall firewall.denizeit.com, leaving no doubt as to what it is.

Most boring companies will use a very simple naming convention, like mail.example.com and firewall.example.com. Although this is highly practical and sensible, you end up telling the outside world a lot of information that should really be kept private. Do you want to tell people what server your firewall is? Or where you keep your extranet? This can be highly useful information to me when a network might be composed of five to ten class C networks, and it can also save me a lot of time searching for a particular service.

Some companies do try a little harder than this and will start to actually come up with some semi-original ideas for naming conventions. The most common that I’ve encountered is a set of names based on the Greek gods. IT system administrators seem to have a fascination with gods. Sadly, it’s very predictable. I have yet to a see a network where Zeus is not the firewall and Hercules was not the most powerful main server, usually the main development server or the mail server.

The best networks I find are the ones where every machine is named sequentially, like ip-202, or each server is named after a random day or month. I like a challenge, needing to dodge and hide, to sneak around and look through shards of jaded glass to find information. But if you’re going to tell me what server is what, I won’t complain.

Time to Get My Hands Dirty

I have decided on a new plan of attack based on what I’m trying to achieve and what I have learned. I know that while the software I’m after will be located inside their network, it won’t be sitting on their Web server, and it probably won’t even be on their FTP server. It will sit very close to the developers. Since earlier versions of the software have been sold on two CDs, chances are the new version will not have been copied onto a different network. Instead, it will most likely have been kept local. This means that there is no point of trying to break into their Web server, since it probably won’t have anything of use to me. This is also where they would expect a hack to take place.

My best bet is getting a username/password for vpn.denizeit.com and attacking the internal development master server, where CD images of the software should be kept. Or I could simply pull the data off a developer’s PC. I’m sure the VPN would be used for employee(s) to work from home and most likely allow connections from any IP. After all, it’s secure and encrypted, so why not allow anyone to connect to it?

Now I don’t know what VPN software they use. It could be a Cisco concentrator, a Microsoft PPTP VPN, a native PPTP of some kind, or something else—I really have no clue. If I try to probe the VPN looking for common ports/traits of each VPN type, I’ll be seen by their firewall. The only way to do this safely is to think like someone who should have access.

I’m going to put myself in the shoes of a fictional employee who works for Denizeit. Her name is Suzy, and she is one of the clerks down at Human Resources on level 2.Tonight, she is trying very hard to get this VPN thing working from home, so she can connect to her computer at work and get to this damn financial report that she is under a lot of pressure to finish on time for Monday. What does she do?

She has no understanding of IP addresses or setting up VPNs, and the instructions that were e-mailed to her when she first learned that she can work from home are now long gone. The information must be available somewhere externally for her to read.

One thing I noted when I ran DNSMAP was the lack of an intranet.denizeit.com. This could be missing for many reasons. It could be called something obscure like intra01, but this is unlikely given the naming convention of all the other servers. They could have the intranet located behind the firewall, making the intranet available only to internal employees. This is possible, but I think that there would be a site or location somewhere on their external network that would show Suzy how to set up a VPN— maybe some after-hours support numbers and general IT support help topics.

My first guess is that they have a section on their main Web site, probably password-protected for internal employees. I guess this because I noticed that there is only one external Web server. Browsing around their Web site, I never saw support.denizeit.com or pressreleases.denizeit.com—just www.denizeit.com. My guess is that they have a Web site hosted with some big hosting company, and they keep everything on this one Web site. I also doubt they would be stupid enough to have their whole intranet live to the outside world. There’s no logical reason for things like complete phonebook listings, private company announcements, and the like to be on an external Web site. But, again, I do think they have some pages to help Suzy here set up her VPN. I come up with a quick mental list of the most obvious

names:

- http://www.denizeit.com/employees
- http://www.denizeit.com/vpn
- http://www.denizeit.com/intranet
- http://www.denizeit.com/internal

Guessing URLs like this, if done correctly, can be a very valuable way of discovering information. A lot of companies will keep log files, for example, stored on a server under the directory logs, or the administration section under /admin, or even their whole intranet under intranet. The trick is to put yourself in the shoes of the person doing it. If you know enough about the systems administrator, predicting him is trivial.

After a few guesses, I find that

http://www.denizeit.com/intranet/login.asp exists. I’m confronted with a front page telling me:

PRIVATE DENIZEIT INC, PLEASE ENTER YOUR DEPARTMENTAL USERNAME AND

PASSWORD

Here’s a login page! It’s kind of scary and my hands start shaking, but this is just what I’m looking for. I wonder what it holds. Okay, it’s time to get an account and find out what’s here … after I get some more coffee. It’s amazing the amount of coffee that can be consumed during a long hacking session. Sometimes, I’ll need to dig thought huge company networks, taking an easy 20 to 40 hours straight. I don’t like to sleep when I’ve broken into a network, so drug use is also common—anything to keep me awake. Looking at this login page, I see it’s rather plain looking: two input boxes, one labeled Username and the other Password, but the absence of anything else tells me a lot.

Login.asp

<form method=post action=check_login.asp>

Username<input type=text name=username>

Passowrd<input type=text name=password>

</form>

I think that when this page was developed, it was developed quickly, and there would probably be 30 lines of code at most in this page. Judging from the text, “PLEASE ENTER YOUR DEPARTMENTAL USERNAME AND PASSWORD,” I get the feeling that there are five to ten logins, one for each department. And if the login is based on each department, maybe different departments see different things? If I were this developer, I would write something like this:

Pseudo Code of check_login.asp

Get username/password from POST.

Connect to a simple sql/access database.

Select rights from table where username = ‘username’ and password =

’password’;

If the password is bad, or username is not found return a page saying

”Bad password” .

Else continue…

Read what rights the user has and display the needed pages.

Easy, really. But now I wonder, was the developer smart enough to parse the user-entered data before he builds his SQL string and executes it?

Injecting SQL is not really a new attack. Although it has been around for a while, developers still write insecure code, and it’s exploitable. Since this page was probably written in 30 minutes on a Monday morning, I highly doubt the developer would have even contemplated SQL injection. I mean what is there to gain? Phone numbers, a few IP addresses, a signup sheet for the company softball team? Hardly a big security breach.

First, I test to make sure the script actually works, I enter a username of sales and password of sales, and I am confronted with a page telling me to check with the head of my department for the current intranet password. Okay, good, it works.

A quick test to see if I can inject SQL data is to enter my username and password as ‘a. The first quote will end the current SQL statement, rewriting it to be:

Select rights from table where username = “a and password = “a;

This should cause the ASP page to fail, since the SQL statement is now invalid. Either an error will be displayed or IIS will simply return an ERROR 500 page. Fingers crossed, I enter my username and password as ‘a, and then click Logon. Bingo!

The Result

Great! It looks like it died when trying to parse my SQL query. Now it’s time to inject some correct SQL statements to see if I can get around this whole password problem.

If I pass the username of a known department (I’ll use sales here, since almost every company always has a Sales department) and a password of “ or ‘1’ = ‘1’, I’ll be creating the following SQL statement:

Select rights from table where username = ‘sales’ and password = “ or

’1’ = ‘1’;

The database will pull the data only if the username sales exists, the password is “ (blank),or 1 is equal to 1.The username sales exists; the password isn’t blank, but 1 does equal 1 (last time I checked). I am greeted with the front page of the intranet, “Welcome Sales Department.”

Getting Inside the VPN

I’m starting to get somewhere. On the left side of the page, I see a navigation menu with the following menus:

Network Status

Bulletin Board

Cafeteria Menu

Support Phone Numbers

Technical FAQ and Help

Logout

A check of the network status shows that there are currently no known issues with the network.The café is serving steak and fries this Friday (ugh, I’m a vegetarian!), and the bulletin board shows that Frank is looking for a new roommate. The support phone numbers listing shows some fairly interesting information:

For all technical support issues, please call Andrew Jacob at 804 1955

Ah, I think to myself, our friend Andrew Jacob, who registered the DNS—he must be the main technical support

guru.

The Technical FAQ and Help page is very interesting though, especially the section about connecting to the VPN from home:

“Denizeit.com allows employees to connect to work from home and access

all work resources. It is suggested that you have at least a cable

Internet connection, as dialup can be very slow.

To set up the VPN connection, click create a new “Network Connection”

under Windows Explorer.

Then select “Create a new connection to my workplace.”

Select the connection type as VPN.

Enter the ip address of the server as vpn.denizeit.com.

Your username will be the same as your email user account or first

letter of your first name, followed by your last name (e.g,

jdoe@denizeit.com username would be jdoe).

Your password is different from your logon password. When your VPN

account is first created, your password will be remoteaccess. We

strongly suggest you contact Andrew Jacob at 702 804 1955 and have

this password changed after the first time you have logged on.

I grab a piece of paper and scribble down “remoteaccess” and the format of the VPN usernames. Then I return to the bulletin board to browse upcoming company events a little more. I’m curious. You never know—if they have some good company events and get a vegetarian menu, I may even think about taking a job here someday. Then again, I probably can make more money stealing software from them.

Now, in a perfect world (for them), I would be no closer to breaking into this network, because all the users would have

changed their passwords after they logged in for the first time. I know for a fact that this isn’t the case. As a whole, mankind is stupid and lazy; if we don’t have to do something, we simply will not. So, I bet that at least one user has not changed his or her VPN password since it was created. I’m limited a little, however, because I still need to know some usernames. I decide to do a little searching around first and build up a list of e-mail accounts, and then try each with the password remoteaccess. What better place to start but their intranet?

The bulletin board has a lot of interoffice communication about general chitchat topics, and I get a list of ten e-mail accounts from various replies. I surf to my favorite search engine (www.google.com) and do a search for @denizeit.com, because I want some more e-mail accounts just to be sure. I also would like to get as many e-mail messages as possible for their IT department, because these guys may have higher access around the network.

My search shows some knowledge base replies from www.denizeit.com/kb/ and a post to a C++ newsgroup, asking a question about advanced 3D matrix transformations. Sounds interesting, although math never really was my strong point. The e-mail account Peter James pjames@denizeit.com, who is asking these questions, probably belongs to a developer—someone who might have access to the software I’m after.

I grab another coffee, sit down with my list of 17 e-mail accounts, and get ready to set up a new VPN connection. I test each account with the password remoteaccess.

Password Fail..

Password Fail..

Password Fail..

Password Fail..

Connection Created OK

Looks like Jamie Macadrane (jmacadrane@denizeit.com) didn’t bother to change her password. I disconnect and try the other usernames. Out of a total of 17 accounts, 4 have the password of remoteaccess, including pjames@denizeit.com.

I am in. An evil smile creeps across my face. I love hacking this way. I haven’t used any known exploits. If their server were patched to the very latest patch level, I would have still gotten in. The weakness I exploited was not in the Web server or network layout, but the people behind the keyboard. A simple way they could have stopped me would have been to have the VPN authenticate off their primary domain server, then simply have each password expire every 30 days. Oh well, I won’t complain.

Finding the Software

My focus, direction, and mindset totally change now. When I was outside the company’s network, I had issues like being detected by firewalls and IDSs. Now that I’m inside the network, these problems are gone, and I can start to relax and really enjoy the hack. Although companies will have a firewall to protect themselves from evil hackers, they will blindly trust anyone inside their network. I have yet to see a network that has a firewall, or solid security, inside the network.

When I was outside the network, I didn’t use port-scanning tools or any other known hacking or security tools. Everything I did looked as innocent as possible. Now that I no longer need to be so cautious, I’ll use some tools to feel around their network.

A quick check of ipconfig shows that I’ve been assigned a DHCP IP address of 192.168.1.200. What I need to do now is find out what the other 252 IP addresses in this network hold. Since this is (so far) a Windows-based network, I’ll take an educated guess on how they will lay out their software development servers.

A Windows server located somewhere internally, probably with a large disk running Microsoft Visual Source Safe. It would have a few Windows file shares, mapping out various sections of code development—probably one for beta code, another for older versions, and maybe a few private shares for developers to share common data among themselves.

A machine for burning CDs, probably a workstation and probably called CDR or BURNER. This would be used to create CDs to be sent to business partners, given to employees to take home, or used for general installations around the office.

I want just the software. If possible, I would rather not need to break into their development server. I just want to get my copy and leave. At this point, most hackers would get greedy and begin to hack every machine, trying to obtain total control. They might think about injecting a backdoor or virus into the developed code, or even just deleting it completely. A mindset like this will lead straight to getting caught. It’s like being at a casino and winning $100.If you’re smart, you’ll leave then. The dummies stick around and try to win more, usually losing it all in the process.

Looking Around

A computer will tell you a lot about itself if you ask it. In the same way that DNS can leak information, WINS (Windows Internet Naming System) can tell you the same, if not more, information. The best way I find to do this is to use fscan (www.foundstone.com) in a passive, resolving mode. What I’m looking for is either a development server or a machine used for creating CDs.

Output of fscan (shortened)

192.168.1.1 coresw1.denizeit.com

192.168.1.2 router.denizeit.com

192.168.1.26 staging

192.168.1.27 dev01

192.168.1.40 97795

192.168.1.41 97825

192.168.1.42 97804

192.168.1.43 97807

192.168.1.44 97818

192.168.1.60 DENIZEIT1

192.168.1.50 HP_4000n

192.168.1.52 CDR42X

192.168.1.102 97173

192.168.1.101 rt2500

192.168.1.100 97725

192.168.1.105 97449

192.168.1.106 192410

192.168.1.138 93066

192.168.1.137 97757

192.168.1.135 LAPTOP1

192.168.1.145 97607

192.168.1.162 laptop2

192.168.1.170 act102801

192.168.1.157 ernie

I cut back a few entries here, but by the looks of it, this is the core network. Seems that everyone is in one subnet, so probably around 200 people work in this company. Not bad.

I guess the four- or five-digit computer names are asset numbers or some kind of tracking numbers. This probably means that all the desktop computers are leased from someone. I also see that my guess of a machine used for burning CDs was not too far off; CDR42X sounds like a safe bet. And dev01 would most likely be their development server. The interesting thing here is the 01. Why call something 01 unless you have 02 or 03? A quick ping of dev02 and dev03 reveals that they are not responding. Probably, their network designers are just leaving room for growth.

Now, I have found my targets. First, I will attack their development server and see if I’m able to connect to any open/null shares. Although I have a VPN account, their Web site told me that this password is different from a user’s login password. This means that I’ll need to connect to any resources as a guest. I will try to get a domain username and password only if I really need to. The key word here is need. I’m not getting paid by the hour, and the software is all I’m after.

I run Windows 2000 on my PC (as well as gentoo Linux). I find that hacking a Windows server is easier if you use Windows. I click Start | Run and type in \\192.168.1.27. This will connect to dev01 and enumerate all publicly available shares if I’m able to connect to the IPC$ (Interprocess Communication) as guest, although it will not show hidden shares (such as c$ or d$).There should be a publicly available share if developers are to use it. Sadly, I see a user login/password prompt. Obviously, I need to be authenticated to connect to the IPC$.

Dang. Well, at least I have the CDR machine left. The thing about CDR machines is that they usually have no security whatsoever. Why bother? It’s just a dumb machine that burns a few CDs, right? What most people don’t realize is that everyone connects to it and copies files to CDR machines. They often contain a wealth of various random data. Most people don’t remove the files they’ve copied to the server. Again, humans are lazy.

I type in \\192.168.1.57 and am greeted with a pop-up box showing three share names: INCOMING, IMAGES, and USER. I now type in \\192.168.1.57\INCOMING. Bingo, I’m in what looks like the dump directory for people to place files to burn. There is everything here from pictures of vacations, random mp3s, and an interesting zip file called Current_website.zip—perhaps a zip of their Web site content, possibly containing some passwords. Most of this looks like general user data, personal information, backups of documents, and so on. After skimming through various files for about half an hour, I decide that this data, although entertaining and informative, isn’t really worth my time.

I bring up the share IMAGES and see the following directories.

DD_3

DD_2.5

DD_2.21

DD_2

DD_GOLD

OfficeXP

Windows XP

COREL DRAW 10

There are also a few other office application directories, but what really catches my eye is the first one, DD_3. It looks like Digital Designer 3 to me. Inside this directory, I see cd1.iso, cd2.iso, and readme.txt.

Readme.txt

Thanks to all who worked on helping make Digital Designer 3 what it is

today.

The license code is: DD3X-1029AZ-AJHZ-JQUE-UIW

This is the multi site license code for unlimited nodes, and is

limited to partners and internal employees ONLY. Do not give this code

out!

Jerald Covark

Head of Software Design

Denizeit Inc

This is wonderful! Obviously, IMAGES holds the CD images of various applications used around the office, including Digital Designer. I remember that when I was checking over their Web site, I saw a list of about 25 business partners. My guess is that this machine was used to create private copies of Digital Designer 3 for them.

The license code is also rather handy. I guess they print this number with the CD when they ship it. This is everything my client needs. I select the files and begin pulling them over the VPN back to my computer. The good thing about the license is that if Denizeit were ever to catch onto the fact that Digital Designer 3 was available prior to its official release, and that every copy was released with the internal private license code, they would first suspect one of their business partners of leaking the CD.

Conclusion

For me, the art of hacking is to have a clear objective and a very clean target. A messy hacker who just wanders around a network looking for trouble will eventually be seen and then caught. There was really only one point in this hack where I could have been seen: during the SQL injection stage of things, when I was breaking into the intranet. A Web log will show that I caused the server to issue a 500 return. Chances are this will go unnoticed.

It’s also important to note that I never even tried to break into the development server. My goal was not to gain source code or maliciously inject a virus. It was simply to steal the company’s most major asset, their software. I would have broken into dev01 only if I had to, in order to gain access to the software.

This network could have been at the latest patch level, with a security administrator sitting on the keyboard every day, and I still would have gotten in. Hacking does not need to involve the latest 0-day exploits and forcefully stumbling around a network. The true hacker is the one who simply uses his mind and exploits small, simple weaknesses in human beings.

I suggest they upgrade to Employee v1.01.