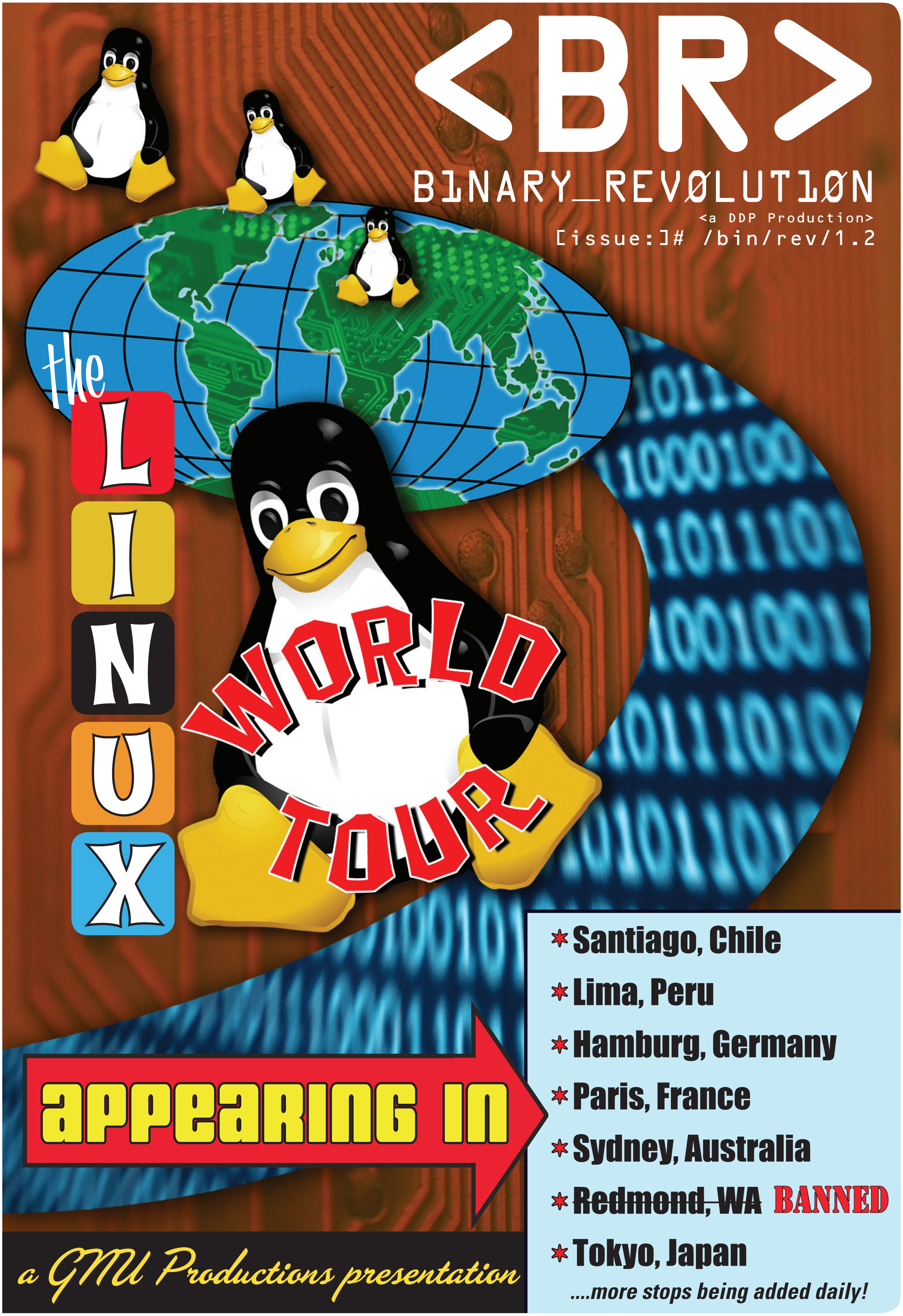
<BR>

B1NARY_REVØLUT1ØN

<a DDP Production>
[issue:]# /bin/rev/1.2



*Santiago, Chile

*Lima, Peru

*Hamburg, Germany

*Paris, France

*Sydney, Australia

*~~Redmond, WA~~ BANNED

*Tokyo, Japan

....more stops being added daily!

<BR>

B1NARY_REVØLUT1ØN

[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
dual_parallel
bland_inquisitor
logan5
notheory
w1nt3rmut3
voodooHAL
Nick84
Rax

/bin/rev/staff/

Editor In Chief: StankDawg
Layout and Design: logan5
Webmasters: bland_inquisitor, StankDawg
Special Thanks: FL4SH

Cover Credits:

Concept: dual_parallel, logan5, and StankDawg
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.

<BR>

B1NARY_REVØLUT1ØN

[a DDP production]

ls /bin/rev/1.2/*.*

Editorial/Introduction

Come one, come all! Ladies and gentlemen, step right up and feast your eyes upon the wonders of the amazing, super-strong and ever-reliable Operating System of the future. Never the same in any location. Each appearance different than the last. You've heard about it, read about it, seen others run scared in its presence. Behold, LINUX, the evolution of Operating Systems! Join the LINUX world tour when it comes to a city or town near you. Get your tickets now.

Touted as the future of computers, and bringing the stable code base of Unix itself, Linux has hit the scene in full force. Once an anomaly used by only the most hardcore geeks and hackers, Linux has now become a viable operating system for home, office, pretty much any and all circumstances. What was once considered the domain of hardcore tech-savvy masters, is now available in a form that is attracting more and more users every day. Where there were relatively few distributions before, now there are literally hundreds. Whatever your needs are, there is probably a distribution for you. Do you want a secure distribution for net commerce? Check out Redhat or SuSe professional distributions. What if you want a general security focused distribution for security officers or other security-minded individuals? Try the new Knoppix STD (security tools distribution). What about a distribution that focuses on reliability? Maybe Debian Linux is for you. And for the crossover crowd, who need to ease into learning a new OS from the world of Microsoft Windows, take a look at distributions like Mandrake or Redhat. If you are worried about the trouble and hassle of installing Linux, or worried about loss of data, you still have options. Before you take the splash and "switch" completely, try the Knoppix distribution of LINUX which boots ENTIRELY FROM THE CD and doesn't touch your current system! Boot Linux only when you want it, or need it, without losing your current OS!

What can Linux do? That is a valid question. Let's do away with some myths about the lack of software for Linux. Linux has ports of many major software applications that are too numerous to mention (you would be surprised). The best, and most common Internet web server is Apache, a Linux application. Photoshop is not ported to Linux, but there is the GIMP (GNU Image Manipulation Program) which is every bit as powerful as Photoshop, if not moreso. Even if there are no direct ports of specific software packages, there are quite a few Linux alternatives. Particularly, there are office packages that offer word processors, spreadsheets, and other office- type programs that are fully compatible with Microsoft Office. You can import Microsoft file formats (.XLS, .DOC, etc...) into these programs easily and save files in common formats as well. The most popular of these office packages is OpenOffice (aka "openoffice.org" for legal reasons) but you can also use Koffice or Star Office as well. Most of these files can then be transferred to other Linux or non-Linux systems and be opened by other applications just as easily as opening any other file. You can take your .DOC file from work, take it home to your Linux box and edit it with Open Office, and then take it back to work the next day without even thinking about it.

Finally, the intangible and immeasurable bonus that you get with Linux, is the fact that the entire OS, from kernel to applications, falls under the GPL (General Public license). According to Linux.org this means that, "...its source code is freely distributed and available to the general public." So, not only do you have this free operating system, and all of these publicly available applications, but you also get the source code. Why is that important? Well, to non-technical people, it might not seem like a big deal, but once you get comfortable with Linux, and the applications that you like, you may have the urge to change or improve the software. Surely you have found an annoyance in software before, but what recourse did you have? Upgrade? Complain to the company who ignores your email? No recourse at all? Well, now you have options. Now, if you have the time and dedication, you can modify the code itself and make your own customized version of the software or OS. With Linux, you have the power and control.

But don't just my word for it. As hackers, who have been using Linux for a long time now, we may be a little bit biased. I can admit that. So lets look at it from an impartial viewpoint. Linux is used by many home users as well as companies and governments. You can buy PCs from walmart and other large retail chains that come preinstalled with different versions of Linux aimed at the home user. But home users aren't the only ones who can take advantage of Linux, so can businesses. Apple recognized the power and stability of Unix and based its latest version of MacOS on a version of Unix! Even Big Blue (IBM) has thrown its weight behind the power of Linux announcing recently that it would ship Linux powered servers. The biggest recent adoption of Linux has come from governments. Germany has implemented Linux in every level of government from federal to local. And they are just one of many. Other countries that have similarly started switching to Linux include: Venezuela, Mexico, Peru, China, Singapore, Australia, and the list goes on. Basically most major countries in the world have either switched, or are considering the switch to Linux. The only noticable absence is the United States.

You can get Linux powered PDAs, laptops, cel phones, home electronics, PC hardware, and almost any gadget you can think of. It is stable and adaptable and can be customized for whatever your needs may be. The hacker community has known this for years, and finally the world is catching on. Join the Linux revolution, and more importantly, join the Binary Revolution! The revolution will be digitized! 0101

StankDawg

$#@%$&! <fsck>

• Last issues ***Perl Corner: 'Watching the Watchers'*** by Nick84 had a formatting error that caused all of his code to be aligned left. The code works, but visually it looked bad. Nick84 wrote much nicer code than we made it look.

HACKING HACKING HACKING 101 HACKING HACKING HACKING

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

Targeting Theory

By StankDawg *(StankDawg@hotmail.com)*

So you want to "hack" a system, eh? But what exactly does that mean? Where exactly does one start to hack into a system? "Teach me how to hack!" Well, this article is fundamental in understanding how to target and attack a system. This article is meant for security professionals and hobbyists to understand the concepts of hacking. This article is also meant for hackers who want to understand the same concept. Even advanced hackers do not fully understand or appreciate the fundamental steps that go on inside the hacker mind. This is a crucial lesson in developing and understanding the mindset of a hacker.

This is a theory topic, meaning it is about understanding a theoretical concept about hacking. This is not a specific detailed account of how to apply a specific exploit or attack on a system. This is on a much higher level. Understanding fundamental concepts such as "targeting theory" will take you much further than a specific one time trick or "how-to" lesson. This is a critical concept to learn and use for all hacking application.

When I say "targeting theory" I am referring to the research, analysis, evaluation, and determination of where the most appropriate place to attack a system is. More simply put: What method and location of attack is going to be the most likely place of gaining access? For "play" hacking, you may simply stumble across a site or a server and just look at what is presented to you. You poke around, perhaps finding some loopholes or logic bugs that allow you some extra access. This is incredibly fun, don't get me wrong, but if you actually want to "root" the system, you need a more analytical approach.

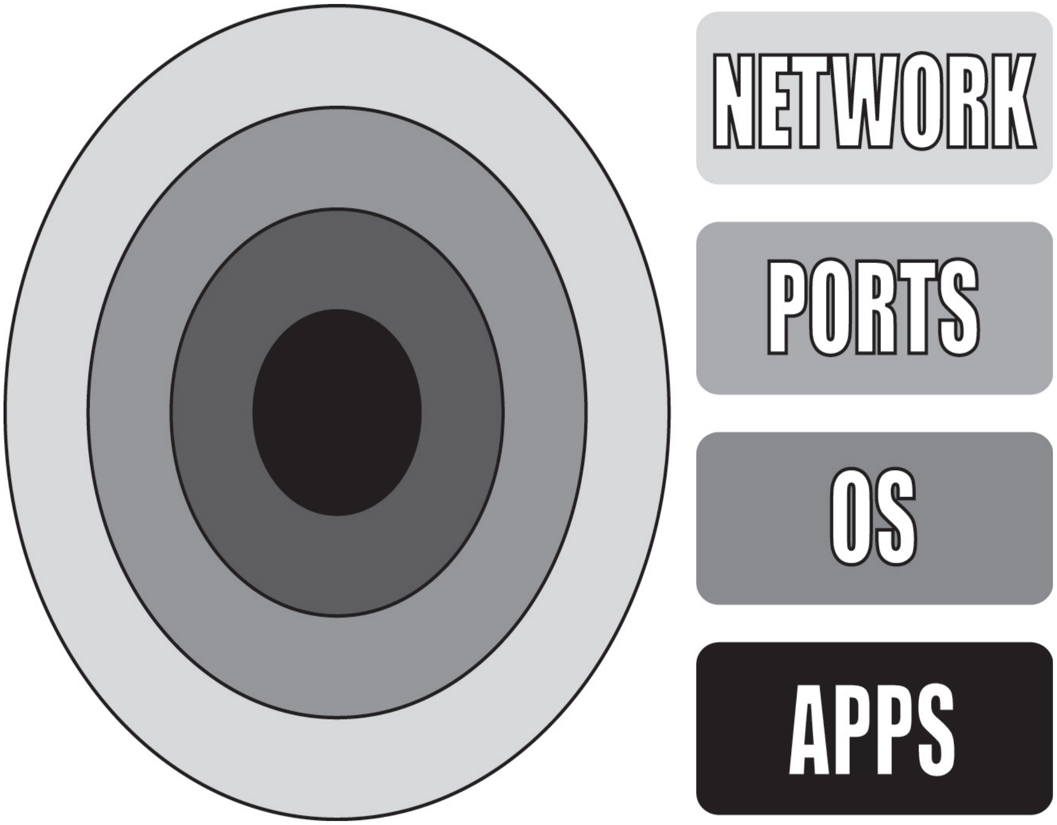
This is when we apply targeting theory. You have a server that you have decided to access (preferably your own for testing purposes, or one that you have permission to test) you apply the techniques of "footprinting" a system to find out everything that you can about that system. Find out the information about how it allows access (internet, dial-up, physical only, etc…). Discover what operating system it runs. See what applications it runs. If it is connected to the internet, what open ports does it have? Basically, any bit of information you can find will help give you an overall understand of the system. Only with this can you begin proper analysis.

Depending on what you find in the results of your footprint analysis, you must then extract the most useful information from the results. Make a list or a chart of the information that you feel is most appropriate. To give a specific example, I would suggest four separate columns of the following: Network, Ports, OS, and, Applications. Take a look at a very small example list that could be using data garnered from footprinting a system. The first article in the ***HACKING 101*** series was about how to footprint a system and was originally published in Binary Revolution magazine issue 1.1 which is available at ***www.binrev.com*** for online ordering.

Network	Ports	OS	Applications
192.168.0.1	21 / FTP	Windows NT 4.0	IIS 3.0
192.168.0.1	80 / HTTP	Windows NT 4.0	IIS 3.0
192.168.0.2	21 / FTP	Linux (Redhat 7.2)	wu-ftpd 2.6.0
192.168.0.2	23 / Telnet	Linux (Redhat 7.2)	??? (unknown)
192.168.0.2	80 / HTTP	Linux (Redhat 7.2)	Apache

From this, you can get an idea of what organizing your data can do in regards to enhancing your clarity. By seeing these facts in a more organized state, you can look for patterns or ideas that may help your overall understanding of the system you are attacking. You can see in this very small example, that there are two systems in this make-believe network. Both seem to be running web servers as well as FTP servers. It also appears that one is running a Telnet server as well. This gives you an idea that perhaps they are comparing server stability between one OS to another. Perhaps they are keeping a secure server separate from a non-secure server. It could be any number of possibilities, but each of them gives you a potential hint on where the best point of attack might be. The fact that the second system, apparently running Redhat 7.2 Linux, is also allowing telnet access. That should draw some focus as well. Perhaps that would be your best way in? But now that you can see all of the server information laid out in front of you, how do you decide where to go next?

In the illustration below, we use a metaphor of a target. Each colored ring, or layer, represents a different point of attack. Any and/or all of these attack points can/may be vulnerable, and you can aim for any part of this target. But there is a "method to the madness" of attacking a system. You probability of success can be increased by focusing your time and point of attack to the locations that are most vulnerable to attack. Also, depending on the focus of your attack, your target may be different than simply "rooting" the system.



Set a goal for your attack. If your goal is to shut down the entire network, or simply sniff traffic, you may be better off if you aim for the outer network ring or the port ring of the target. Breaking into a vulnerable application at the center of the target may not be the best way to accomplish that goal. If the goal is to plant a virus, or a Trojan, it may be better to aim closer to the center of the target at the OS ring, or the application ring. There is an old saying that goes, "the right tool for the right job". In this circumstance, it may be better stated as, "the right form of attack for the right job". Generally, the further you are away from the target, the less power you will have. In most instances, these attacks are easier. The closer you are to the center of this target, the more power, access, and potential for damage you will encounter.

Network: The network ring represents more of a hardware target. If your goal is to DoS (create a "Denial of Service") an Internet site, or jam the phone lines of a radio station, you will most likely be interested in focusing on the outer edge of the target. Generally the worst damage you can do by attacking the network ring is preventing normal operation of the target. The reason it sits on the outer edge is two-fold:

1) Your attack will be very vague, but have limited access.
2) Attacking a hardware target is fairly easy.

Developing DoS tools is very difficult, but using them is very simple. They are readily accessible on the internet. By learning as much as you can about the system you want to attack, you can choose the right tools for the job. If your goal is DoS, you can find clients that will accomplish that goal. If you are trying to jam phone lines for a contest, you can use your own knowledge about the system to do this as well. If you know they have 10 lines, you can get an old wardialer program out and start hammering away at all 10 lines at once. Network level attacks can sometimes be very low-tech.

Ports: Perhaps your goal is slightly deeper than preventing access to a computer by physical means. Perhaps you would prefer to prevent access by blocking the ports on a system. Or perhaps you want to delete and/or deface the company's web site. In these cases, you may be concentrating more on hammering at open ports on the system. This is similar to a network attack in some ways, and technically is a part of networking and network interface in general. Because the methods of attack are slightly different, I feel that network interfaces, in the forms of open ports, is a target that stands separate from the physical network. You see in the spreadsheet on Page 4 that one of these servers has 3 open ports: 21, 23, and 80.

These ports are open to the public, although access may be restricted by the use of passwords or other forms of authentication.

When attacking these open ports, the best tools to use will be different password cracking tools. If you are trying to gain access to a password protected website, for example, you may use a password cracker used specifically for web page forms. These programs use brute force to continually post username and password combinations into the page trying to gain access. If you are faced with an FTP login prompt, you can try a slightly different password tool. This is also the place where we hear the term "Social Engineering" quite frequently. The attacker calls up unsuspecting employees and attempts to extract information from them over the phone that they can use to help infiltrate the system. A dim-witted manager or a disgruntled clerk may be easily fooled into giving up their username and password. With this, it is a matter of locating and modifying the target to accomplish the goal.

OS: The operating system is the most vague to define as a point of attack. It pretty much goes without saying that if you gain access to the overall system security, you will truly have root of the system. It is for this reason that it is close to the center of the target. To this end, you need to do some true investigation of the OS that you are trying to attack. Find out the exact versions of the software, right down to the kernel. Find out what the default settings are. Does it install a firewall by default? What about authentication? Does it have a guest account or administrator account? All of these things are related more to the Operating System.

The direct way into an operating system is to find out if there are any default openings to attempt to gain access. If there is a "guest" account with an easy to guess password, then you instantly have access to the system. Unfortunately, this may not be enough access to get what you want. The local security settings for that system may prevent the guest account from editing files or accessing system commands. You may need to find an account with more permission to be able to "tag" the system with a small text file that says "DDP was here" or whatever other goal you had in mind.

Apps: The center of the target, and the most precise way to attack a system with the goal of controlling it, is the specific individual application itself. In footprinting the system, you will get an idea of almost every software application that it runs. This is where system security is hardest to maintain from an administrator standpoint. This also means that it is the most likely source of vulnerabilities from an attacker standpoint.

As an administrator, you may have to maintain hundreds, or even thousands of systems. Even if you establish a standard desktop environment, and even if you control the installation and implementation of these desktops, you will still have the most difficult task of reasonably maintaining all of these systems. You may roll out 300 systems with your OS of choice with the latest service packs, security updates, and virus protection software. The bottom line is that once they arrive at their destination, you can no longer control the system. The users can, and frequently do, change settings and configuration. No matter what steps you take to restore these settings, they will continue to change them. Users do not want to hear about your security concerns; they simply want to get their jobs done. That means making whatever changes to their system that they want; changes that may include installing unapproved software.

Hackers live for finding these "rogue" applications. Sometimes, they even implement these applications themselves in the form of "Trojan horse" programs which plants an application on the target system. As new vulnerabilities are found and released, how long will it take the administrators of a network to roll out the updates? And what can the administrators do about applications that they do not know exist? The answer is nothing! And by creating a detailed footprint of the target system, such as the example explained earlier, hackers will find the vulnerabilities before the administrators. Find and exploit the vulnerabilities on these individual applications and you will have the greatest chance of gaining the most access on the system that is possible. This is what hacking theory is all about.

In summary, there are some important facts that need to be emphasized. Hackers know that if they gain access to one system, the rest will fall. Your secured network is only as secure as its weakest link. The experienced hacker will have an understanding about your network that will probably exceed that of your own employees. This is one of the overall lessons in this article. Analysis, organization, and application of hacking fundamentals will always give hackers the upper hand. Hackers that flail away without organization and proper technique will be less likely to succeed. Hack ***strong***, hack ***fast***, and hack ***smart***!

Computing Number Systems

BY: feend

feend [at] binrev.com

INTRODUCTION

My intentions for writing this article are twofold. Firstly, it is meant as an introduction to number bases (specifically those used in computing). Secondly, this text should be able to serve as a reference when trying to recall the methods of converting number bases and doing arithmetic with those bases specific to computing.

Numbers and their bases are of course the essence of math as well as computers. When I refer to a base I'm talking about the number of symbols used to represent a quantity before there is a shift in position. Let's take a look at the system you and I are most familiar with.

DECIMAL

Decimal has a base of ten meaning 10 values can be represented (numerals 0-9) before the magnitude is increased. If this doesn't immediately make sense maybe this example will clear things up:

```
157 = 1x10^2 + 5x10^1 + 7x10^0
        ------      ------      ------
      hundreds     tens        ones
```

The number 9 is the highest number that can be used before another symbol is necessary to show a higher magnitude. This problem is solved by using the same numerals placed at different positions. With the base 10 system each of these positions are incrementing powers of 10. The powers of 10 increase from right to left starting at 0 and going to *n*th power. If you wanted to represent fractional values then those would be represented by decreasing powers of 10 from left to right:

```
157.146 = 1x10^2 + 5x10^1 + 7x10^0 + 1x10^-1 + 4x10^-2 + 6x10^-3
```

BINARY

Although the decimal system is a convenient way for people to represent numbers it is inconvenient for computers. To understand why this is the case you must first be familiar with the terms analog and digital. When you are describing an analog system (ex: the temperature as shown by a thermometer) it can have a continuous set of values. In such a system determining the exactness of a value (is it 73.3 degrees Fahrenheit or 73.367?) can be difficult. Luckily most people don't care about anything past the decimal point and an analog system is fine when used to describe the temperature. Computers don't have this luxury and must have exact precision if they are expected to do anything meaningful with the data supplied. A way to get this exact precision is to use a digital system. This system only uses a finite set of values (ex: a stoplight - green, yellow, and red). Typically computers use the digital system more commonly known as binary. This is because there are only two possible states HIGH and LOW. I'll get into the whys behind binary in a different article. For now let's focus onthe system itself.

```
256    128    64    32    16     8     4     2     1
---    ---    --    --    --     -     -     -     -
2^8    2^7   2^6   2^5   2^4   2^3   2^2   2^1   2^0
______________________________________________________
```

This is the standard 8 bit binary table. The term bit is just shorthand for binary digit. The binary system is almost exactly like the decimal. Numerals are placed at different positions to show magnitude, powers increase from right to left starting at 0 and going to the *n*th power, fractions are decreasing powers starting at -1 and going from left to right, etc. There are only two differences. Binary uses a base of 2 and the only numerals available to fill that base are 1 and 0.

COUNTING IN BINARY

Counting in binary can sometimes be tricky so you need to remember that the exponent is the maximum magnitude that can be expressed INCLUDING ZERO! Look at the example below (note: The b is the suffix used to denote a binary value).

16 = 10000b ...Hmm but why? Well let's look.

8	4	2	1	
2^3	2^2	2^1	2^0	
1	1	1	1	or 8 + 4 + 2 + 1 = 15

A 4 bit number can only express a decimal number in the range of (0-15). In order to increase the range we must increase the power.

16	8	4	2	1	
2^4	2^3	2^2	2^1	2^0	
1	0	0	0	0	gives you 16 + 0 + 0 + 0 + 0 = 16

There is an easy way to always know the range of a binary number. This can be accomplished with the formula 2^*n*-1

2^6-1 = 63
making the range (0-63)

CONVERTING FROM BINARY TO DECIMAL

You have already seen an example of this above. You simply add the position values of the binary numbers. Let's use an earlier example.

256	128	64	32	16	8	4	2	1
2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
0	1	0	0	1	1	1	0	1

010011101b = 157 or
128 + 16 + 8 + 4 + 1 = 157

CONVERTING FROM DECIMAL TO BINARY

Method 1: Method one is pretty similar to converting from binary to decimal. You just do the reverse and break up the decimal number into the values at each binary position.

Method 2: I like method 2 better because it takes all the guesswork out of things. Commonly referred to as the "repeated division by 2" method. To be familiar with this you need to be familiar with the terms MSB (Most Significant Bit) and LSB (Least Significant Bit). The MSB is the value at the leftmost position and the LSB is at the rightmost position.

Method 2 is based on remainders. What you do is keep dividing the whole number quotient by 2 and if you get a remainder of .5 a 1 goes at that position. If you get a whole number then a zero goes at that position. Divisions start from the LSB to the MSB. This process is hard to explain so let me clear it up with an example using trusty 157.

```
157/2 = 78.5 -------------------------┐
78/2 = 39 -------------------------┐  ¦
39/2 = 19.5 -------------------┐   ¦  ¦
19/2 = 9.5 -----------------┐  ¦   ¦  ¦
9/2 = 4.5 ---------------┐  ¦  ¦   ¦  ¦
4/2 = 2  -----------┐    ¦  ¦  ¦   ¦  ¦
2/2 = 1  --------┐  ¦    ¦  ¦  ¦   ¦  ¦
1/2 = .5 ---┐    ¦  ¦    ¦  ¦  ¦   ¦  ¦
            1    0  0    1  1  1   0  1
```

You can see that the result is the same as before (since leading 0's don't matter). You may be wondering about what happens if you further divide .5/2. Well you get .25 and even though it is a remainder it is less than .5 making a 0 at that position.

HEXADECIMAL

Hmm but it only takes 3 numerals to express the number 255 in decimal, in binary it takes 8. Sucks huh? You think that is bad try something like 125096 in binary. OK I'll do it for you.

125096 = 11110100010101000b

Yikes. In walks the hexadecimal number system making things MUCH easier. Hexadecimal uses a base of 16. Hmm with a base of 16 won't I need some more numerals? *In Ed McMahon voice* YOU ARE CORRECT SIR! To get the additional numerals hexadecimal borrows from the decimal system and the English alphabet (note: The h suffix is just to denote a hexadecimal value).

-First 10 Numerals-

0 1 2 3 4 5 6 7 8 9

-The Remaining 6-

A B C D E F **(usually represented in their uppercase form)**

THERE WE HAVE IT A BASE OF 16!

Here's a little chart that you want to memorize as you will see these again and again:

0 = 0000b = 0h	**1 = 0001b = 1h**	**2 = 0010b = 2h**	**3 = 0011b = 3h**
4 = 0100b = 4h	**5 = 0101b = 5h**	**6 = 0110b = 6h**	**7 = 0111b = 7h**
8 = 1000b = 8h	**9 = 1001b = 9h**	**10 = 1010b = Ah**	**11 = 1011b = Bh**
12 = 1100b = Ch	**13 = 1101b = Dh**	**14 = 1110b = Eh**	**15 = 1111b = Fh**

CONVERTING FROM BINARY TO HEXADECIMAL

Now you really get to see the beauty of the hexadecimal system. Remember this example?

125096 = 11110100010101000b

Well here it is converted.

```
1   1110   1000   1010   1000
-   ----   ----   ----   ----
1    E      8      A      8
```

125096 = 1E8A8h

Much easier to look at that way. When converting from binary to hexadecimal you first want to break the binary value up into groups of four or less. From there converting is just a simple matter of knowing the above chart.

CONVERTING FROM HEXADECIMAL TO BINARY

As you can probably guess it's just the reverse of what was just done but for clarifications sake.

125096 = 1E8A8h

```
1        E        8        A        8
-        -        -        -        -
1       1110     1000     1010     1000
```

125096 = 11110100010101000b

INSECURITIES IN MY CAFE CUP

by: vooduHAL
VOODUHAL@YAHOO.COM

FIRST OFF, this is not designed to be a technical article by any means, and I don't condone breaking any type of security for personal gain. With that said, most all of us have had the joy of being flat broke at one point in our lives, so this article is here to help those unfortunate few who just have to have one more frag understand how to break the very simple security on this internet cafe management software. I have made this vulnerability known not only to the owners of the cafes I have been to, but also the developers of this software who don't seem to care.

My Cafe Cup ***(www.mycafecup.com)*** is designed to be an efficient and inexpensive internet cafe management software. In reality it is little more than a screen saver that connects to a central server to retrieve user account information. I have seen about 10 internet cafes in the south east US using this software, all with the same general configuration. All but 2 of these cafes were running it on MS Windows 98 workstations. Once you obtain a valid user ID and password from the mostly inept person at the counter you log into what looks a lot like a screen saver with a log in prompt. That is basically what it is. Try running notepad or any other application then log out.

Once you log back in you will see that your application is still sitting there waiting for input. At first I thought it was saving session information for later continuation of your session, but after obtaining a second user ID I soon realized that this wasn't the case. It really is just running a foreground application that fills the screen and blocks the use of *CTRL-ALT-DELETE.* The first thing you'll notice is that you have complete control of the workstation. The first thougtht I had was to install some type of quick access trojan that would dump me to a desktop.

Sounded like a good idea, but I soon realized that the place I was at used Adaptec Go Back at every restart so that idea was out. Next I tried to resize and move the main window using the Win32 API. This worked but some mechanism on a timer seemed to check the main window periodically and eventially restore it after about 15 seconds. Method 2 was out. Then I started thinking since I can control the size of the window maybe I can control other parts of it. Thats when it hit me. What if I send the the

WM_CLOSE message? I had my trusty laptop with me at the time, so I fired up Win2k and started up VC++. After about 15 minutes I figured out a method of attack. I realized that the easiest way to get this to work would be to log in and then run my app that would send the WM_QUIT message. Hmmm...

Now how would I get the window handle of a window that was now closed. Ah, what if I just started the app with a 15 second delay, log out, grab the handle of the top window, and then send the WM_CLOSE message. Bingo. The cafe cup client happily obliged and closed leaving me with a desktop and no time was being taken off of my account. Even though my job was done, I thought what happens if I don't even have the money to buy a $3/hr block. No problem. I went home and burned my app to a CD with an auto-start script. I went back to make sure and sure enough. The workstation autostarted my CD even with the login screen up, which quickly closed. Now I never have to pay to check my e-mail (joke). The only caveat to this method is that unless you are in a cafe with 50+ machines, the admin can easily see that a workstation is not on for some reason, but if they are like most places I've seen, there will be 3-4 machines that have a problem of some sort that aren't on anyway so that helps you get away with it a little easier.

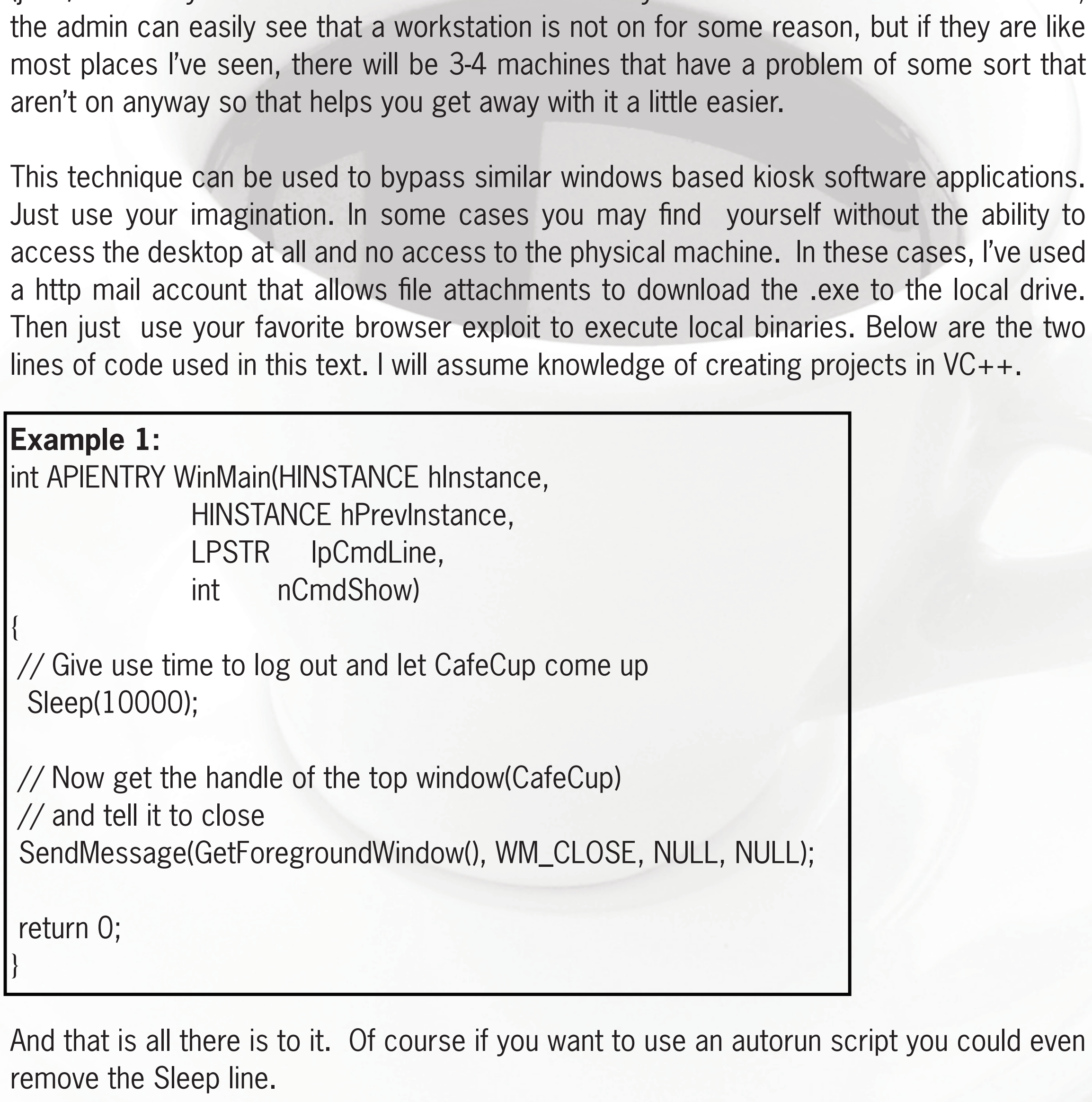
This technique can be used to bypass similar windows based kiosk software applications. Just use your imagination. In some cases you may find yourself without the ability to access the desktop at all and no access to the physical machine. In these cases, I've used a http mail account that allows file attachments to download the .exe to the local drive. Then just use your favorite browser exploit to execute local binaries. Below are the two lines of code used in this text. I will assume knowledge of creating projects in VC++.

Example 1:

```
int APIENTRY WinMain(HINSTANCE hInstance,
                     HINSTANCE hPrevInstance,
                     LPSTR     lpCmdLine,
                     int       nCmdShow)
{
 // Give use time to log out and let CafeCup come up
  Sleep(10000);

 // Now get the handle of the top window(CafeCup)
 // and tell it to close
 SendMessage(GetForegroundWindow(), WM_CLOSE, NULL, NULL);

 return 0;
}
```

And that is all there is to it. Of course if you want to use an autorun script you could even remove the Sleep line.

Happy gaming.



DISCLAIMER

THE FOLLOWING IS INTENDED FOR INFORMATIONAL PURPOSES ONLY. DO NOT USE ANY OF THE FOLLOWING INFORMATION FOR ANY REASON OTHER THAN RESEARCH OR A GOOD LAUGH. BY CONTINUING TO READ THE FOLLOWING YOU AGREE NOT TO HOLD MYSELF RESPONSIBLE IF YOU GET ARRESTED OR BREAK THE LAW AND YOU AGREE THAT IT IS NOT MY FAULT IF YOUR GRANDMA GETS SEIZED BY THE SECRET SERVICE OR BLOWS UP OR YOU SUDDENLY FIND GOD OR ALLAH OR WHATEVER AND DECIDE TO USE THE INFORMATION FOLLOWING THIS DISCLAIMER TO BLOW UP SOME BUILDING. IN SHORT, ONCE FINISHED READING THIS, JUST FORGET EVERYTHING YOU READ. YOU HAVE BEEN WARNED. PUNK.

cloning phones on the sprint pcs network

by psypete

"AHEM".

Settle down class. It's time for Mr. PsyPete to school y'all on some stuff that would constitute schoolins. Here's a basic overview of cell cloning on today's Sprint network.

First, a review. Class, who knows what cellphone cloning is? Anybody? Bueler? Well class, cellphone cloning is the process of putting service from one cellphone onto another cellphone. It's the same thing you do when you buy a new phone and they put your existing service onto it. Only cloning does it a bit differently because sometimes people that clone cellphones don't exactly work at Sprint :]. Really though sprint should have no qualms with you doing this to your own legally purchased phone and service plan considering they tell you how to do it (which i'll go over in a bit). Keep in mind that I Am Not A Lawyer and I do not know for a fact what Sprint does or does not recommend or condone on their network with their phones and/or their service, so for our purposes lets say don't ever ever ever try this at home kids. Also, I'm gonna end the whole teacher shtick right about here, considering it's stupid.

First, the terminology:

ESN: ***Electronic Serial Number.*** A 32-bit number assigned by the mobile station manufacturer, uniquely identifying the mobile station equipment.

NAM: ***Numeric Assignment Module***. It is information that is obtained from the service provider and programmed into the phone at the time of sale (ortime of cloning). It includes the MIN, SID, and GID.

SID: ***System IDentification***, a number assigned to each cellular carrier in a different region.

GID: ***Group ID***. This is used to indicate which country the phone is registered in. It is a pretty useless parameter, duplicating the function of the SID, but MUST be entered correctly. North America's GID is 10.

MIN/MDN: Mobile Identification Number. In the United States, this is simply the telephone number. In other countries, the first four digits of the phone number often have to be translated into the correct MIN code.

IMSI: ***International Mobile Station Identity***. A method of identifying stations in the land mobile service.

IMSI_S: A 10-digit number derived from the IMSI. IMSI_S generally corresponds to the last 10 digits of the IMSI.

With that under your belt, you're ready to start the overview. Cloning a cellphone on a sprint network is very simple.

Step 1: Get the ESN and optionally the MIN if you don't get it in steps 3 to 4.
Step 2: Get it associated with the account
Step 3: Get the MSL and SID
Step 4: Enter the programming and enter your SID and MIN into NAM1.
Step 5: Wait 2 hours and you're done

Assuming you have your own cellphone and service plan with Sprint, all this is quite easy and AFAIK legal, but IANAL and YMMV. We'll use the "legal" method here which involves calling up Sprint PCS Customer Care and essentially "activating" the phone. There are three methods of calling up Sprint to do this: Go into your local Sprint store and use the nearest Customer Solutions phone; dial *2 Talk on your Sprint PCS phone; call **1-888-715-4588** from your home or office phone. First they'll ask you information for a credit check. This will probably need to be the same information on the Sprint PCS account. Social security number, name, address, etc. But not a credit card number. Once you get past this phase you're home free.

Step 1: Now the customer service rep will ask for the ESN. The ESN of the phone is on the front cover of the box of the cellphone (usually a sticker stuck on for pricing) or on the back of the phone. So when you see a bunch of boxes of phones at walmart sitting out with the ESN exposed... well I don't really need to explain, do I? :-) The MIN is the phone number you're cloning, and most likely you'll have this already, but if you don't the customer service rep will tell it to you.

Step 2: Give the customer service rep the ESN. They'll then associate the ESN with the Sprint account so when you use the cloned phone you aren't red flagged (immediately). I'm not positive but I think cloning would work without this step. I haven't done this of course (heh heh) so I wouldn't know. ;)

Step 3: Now the customer service rep will give you the MSL and SID numbers to enter into the programming in the appropriate places. They're pretty clearly labeled in the NAM programming. The MSL is a number unique to the provider and your phone which is supposedly derived from an algorithm on some general information about the phone, though from what i've seen of the Sprint PCS activation website it looks almost like a static number. Anyway. The SID is a number which identifies your home cell region. I think this was instituted for roaming purposes, but with Sprint's current PCS network you have free roaming anywhere in the country as long as you're on a Sprint PCS network, so it shouldn't matter for roaming. You still need it though. There are old lists on the web if you somehow don't have yours. You could also ask a friend with Sprint PCS service to see the SID on their phone. Not all carriers seem to use the same SID, it seems, as I've compared SIDs on more than one network for the same general area and they weren't the same numbers. But who cares, you just need the damn number. The customer service reps have to give you the MSL and SID so don't accept their story if they tell you that you can't have it. Social engineering the local Sprint store's dumbass employees is another way to get it.

Step 4: Now comes the programming. There are a few fields to put in that the customer service reps don't tell you about, and i'll go over those briefly. On a Sanyo SCP-4500 or 4900, to get into the programming, go to the main screen where you would dial in a phone number to call. Enter in "##", then the MSL. Then press the Menu or OK button. Click "Save Phone #". It will then shoot you into programing. The first field on a 4900 will be the ESN; you should already have this, so ignore it, press menu or ok or whatever to go to the next item. It should be the Mobile Number (MDN), which is the phone number to clone. Enter it in here. A sample would be "2025551111". Go to the next menu item. It should say IMNI_S. Enter in the same number you entered before. Go to the next menu item. It should be the mobile country code. This is essentially the GID. The default should be 10 for the US of A, but I've found most Sprint phones uses 310. The next menu item should be the network code. Make this "00". If it brings you back to the programming screen when you go to the next item, you're done; just press exit and it'll reboot and you're done. If it asked you for the SID, enter in the SID. Some Sprint phones don't ask for the SID so i'll assume they just download it when they get service.

Step 5: Now most of the customer service reps say it'll take up to two hours to get service. In fact, it can take up to 2 business days. It all depends. **KEEP THE PHONE ON AND DON'T TOUCH IT!** The phone needs to stay on to get some weird crap sent to it, and many activations get screwed up by skipping this vital step.

There are 2 other ways to activate Sprint phones which are pretty simple. The first I can't recall intimately, but essentially you call up some Sprint 800 number and you get a set of automated prompts. You go into some activation field and the nice machine will ask you for your ESN and such, and give you the programming information. AFAIK, you don't need the background information for this step. The other way (which I used to program my new Sprint PCS phone I purchased) is to use the Sprint PCS Activation website at ***http://activate.sprintpcs.com***. There you need the following information: your current PCS phone number, and your account password. You can retrieve the account password by going to: ***https://manage1.sprintpcs.com/Manage?action=doLostPassword*** and entering in your Sprint PCS phone number. This will send a little text message to the "old" phone containing the password to use on this website. With phone number and password in hand, select "I currently have PCS Service and want to add another phone or replace my current phone with a new one." and click **Start** on the form. This will bring you to a page asking for the phone number and password. Enter them in and select "Replace a current PCS phone with a new one without making any changes to your phone number".

Click “OK”. This should let you into a new page asking if you’re programming it for the first time or if you messed up and need to do it again. Go into the part where you messed up and need to do it again. This is where the MSL comes in handy. You see, the SPC is used the first time the phone is programmed. It’s the same as the MSL, only it won’t work after the first use. That way customers could program the phone once for activation. Well, they’ll give you the MSL here in case you messed up the programming with the SPC, and this MSL will work from here on out each time you want to access the programming. Lucky you :-)

Just follow the website through it’s instructions and everything should work out nicely.

Quick shout out to all the Florida 2600 meetings (‘specially 954) and the following websites for their helpful tips:

http://www.bridog.net/cellular/
http://www.google.com/

Also a big thanks to Sprint’s customer service reps for being so good about giving out information.

cin<<feedback

Question:
What would be the best flavour of Linux/Unix for a complete beginner to start with?

Boo!

Answer:
I would recommend either Mandrake Linux, or RedHat Linux. I haven't been able to test the latest version of either of them, but I've tried Red Hat 8.0 (latest 9.0) and Mandrake 9.0 (latest 9.1). Mandrake is easy to use, but there aren't that many books for it, so Red Hat might be the best. At least it's the most all around system.

Cr4X

Question:
I need help, i'm not much of a smart guy when it comes to this topic, been using windows my whole life. want to get a free bsd going in my room. I'm not sure how to install it. Help!

need help

Answer:
http://www.bsdforums.com
http //www.daemonnews.org
http://www.freebsdportal.com

Psychopuppy

Question:
So I called up the closest radioshack and asked if they had any tone dialers, the lady was extremely helpful with her "What the hell is that?!" so I just hung up and called the next closest one, the dude said they quit carrying them. I still want to pick one up, so I wanted to get your suggestions of where to buy one or order one online.

tilded

Answer:
*Well legally, you can get tone dialers, they do exist. ebay is the best bet, but pricey. you will rarely find them at radio shacks, unless its one that is rarely trafficked (dual told me this one). It *IS*, however, illegal to have a redbox, aka a device which reproduces nickel, dime, and quarter tones that were and in some cases still are used to complete phone calls from payphones, but having a tone dialer to produce DTMF tones is not illegal.*

-w1nt3rmut3

Question:
I've gotten myself a new computer with a GeForce FX 5200 Ultra from Creative Labs. I've bought a 150Mhz video splitter, allowing me to install two monitors, wich to a certain degree works; since my computer thinks I only have one monitor it shows the exact same thing on both monitors, wich I can't use. So my question is, how do I install the secound monitor?

Cr4X

Answer:
I have a very similar setup with my GForce 4 at home. The DVI connector on your card is meant for digital monitors (such as flat panel monitors). If you have one of those you will not need the adapter. Todays video cards are capable of outputting seperate video signals which Windows can use for multiple monitor setups. Just a couple years ago you would have to have had seperate video cards to do this little trick. I believe with Win XP you can have at least 8 monitors if not more. The card that is pictured also has an S-Video out which can be used to connect it to a TV. While I have not tried or done the research, it may be possible to connect a third monitor to this output with the right adapter.

Since I am at work now, I will not be able to remember all the specifics. It looks like you have the right adapter. You will also need to make sure you have the latest nVidia drivers. To enable multilple monitors in XP, you will need to right click on your desktop and select properties and then choose the settings tab. You should see two boxes with numbers inside them. One may be grayed out. Select the grayed out one and then check the box that says 'Extend my Windows destop onto this monitor' and then choose apply. To get it working properly you may have to click on the advanced button and make additional changes within the nVidia driver settings.

fl4sh

Question:
I've heard of something called an extender, what is it???

Phreakblaze

Answer:
From the alt.phreaking FAQ 2.2.1 What is an extender? Unlike most systems exploited by phreaks, a WATS extender is designed to be used for making phone calls without directly billing the caller. WATS extenders are 800 numbers connected to bulk rate billed telephone lines and guarded by a pass code (usually a VERY LONG one). "950s" are another common form of extender. The most common incarnation of extenders today is the dialup used for prepaid phonecards. Be warned: extenders VERY often utilize real time ANI, and do not react well to abuse. These things are dangerous and should be treated with care."

ntheory

Question:
I read a text about how redboxing is over. It confused me. Is it really over?

crash

Answer:
Red boxing is dead in most inter-citys if you live out in the middle of no where you might be able to box a few calls. Theres also quite a few payphone companys that mute the handset until you input your money.

My advice is don't waste your time with red boxing. Learn about cocots and other ways of going about getting free phone calls.
NOTE: *Phreaking is not all about getting free calls.*

Ic0n

Question:
I been studing phreaking for short time now and was interested in learning more. What is op-diverting?

Phax

Answer:
*For op diverting, check "Basic Phreak Fun" at **http://www.oldskoolphreak.com/tfiles/basic_phreak.txt** and "How to Op Divert Using Your Local Op" at **http://www.oldskoolphreak.com/tfiles/opdivert.txt***

dual_parallel

Question:
I know a little bit about phreaking but not much. I've read some files about 4 years and thought that using Colored Boxes was dead. But now some of the stuff I'm reading im not so sure. i know some things like cheating cocots will be around for a while but what about Blue and Red boxes do they still work?

unknown_entity

Answer:
On modern switching equipment a blue box won't do anything. The trunks don't respond to 2600hz anymore because all the signaling is out-of-band.

Red boxes still work in places. Not sure how worth it it is to build one nowadays though. At least around me I'd have to search pretty hard to find a payphone that worked with it.

The only tone emitting colored box that I know of that still works is the orange box. Unfortunately all of them are done in software because it's a lot more than just a few simple tones. It's essentially a Bell 202 (correct me if I'm wrong here) modem-ish device to spoof caller ID. It's probably better left to software so you don't have to create a real world interface for it.

If you really wanted to try blue boxing you could go to Nantes, Quebec though. They still have an old crossbar or step switch. But I think it's the last in North America. Check out Evan Doorbell's recordings to see what he says about Nantes. He's a famous phone phreak and has lots of recordings of what he calls "phone trips". They're definitely a must listen if you're a phreak.

ntheory

Question:
I was just wondering what the difference between a real UNIX system, and a Linux system is. Thanks in advance.

Cr4X

Answer:
That's pretty vague... If you expect a list of differences, it could get phenomenally long. The basic difference is that UNIX came first. Different flavors of UNIX came along afterwards. Linux was one of those flavors. It was open source, and freely distributable, which is why it become so popular.

As far as specific differences, it is a little tough. The apps are compiled differently for each flavor of unix and therefore may have different switches and ways of working. These are APPLICATION differences, however, and not KERNEL differences. Kernel differences are the important ones and they are on a very technical level.

StankDawg

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

StankDawg

case MODeLING

OR: How i Learned to Stop Worrying and Love My Beige Computer

by logan5 (logan5@oldskoolphreak.com)

AS far back as I can remember, I've loved building scale models: aircraft, armor, ships, sci-fi stuff.....I've built them all. I've even scratch-built a few spacecraft of my own design using spare model parts, sheets of plastic and odd & ends from around the house. I love the challenge of recreating a duplicate of something in miniature, and learning new construction and painting techniques.

Also as far back as I can remember, I've hated beige computers. I loathe their blandness. Even before fruit-flavored iMacs, and grey and silver G4s started showing up, I'd always wanted to do something, **ANYTHING** to spare myself from having to look at another boring, beige computer. But, since I've always had to shell out good, hard-earned Chlorophyll George's to buy my computers, hacking one up, painting it in German splinter camouflage with chrome racing stripes, Iron Crosses and Betty Page stickers was just something I could never bring myself to do.

But thanks to the modern miracle of dumpster diving, one is able to (with a bit of luck) haul a functional, usable (albeit, probably outdated and obsolete) computer from the brink of destruction and resurrect it in thine own vision, and not have to worry about defacing your hard earned technology investments. So why not have some fun and experiment? After all, it's just garbage, right?

Being a longtime Mac user, I'm quite happy with the way my G4s and iMacs look, and I have no desire to change their appearance. So, when I hauled an old (but perfectly functioning) IBM PC300 PL (Fig. 1) from the dumpster of the state teacher's union HQ building, I knew that it's beige blandness would clash with the rest of the tech in my home. Then the otherwise dim light bulb that seems to hover over my head lit up: this would be the perfect opportunity to take some of the skills and techniques I use in building models and use them to turn this boring, beige desktop computer into something really cool. If anything, it would give me a unique and visually interesting machine to play with Linux on. Thus began my attempt to splice the worlds of model building and computing into one beautiful monstrosity.

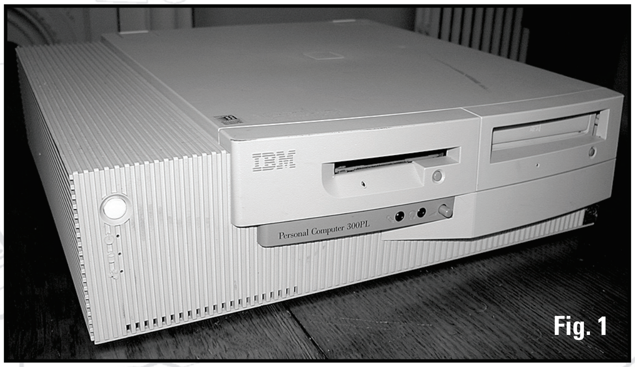


Fig. 1

With this article I hope to show you how, with a trip to the hobby shop and someplace like a hardware store or Home Depot, some spare model parts, and other junk lying about the house, you can take a boring old beige box and turn it into something to make any Borg Drone turn Luftwaffe Dunklegrün with envy.

Due to the internal design of the subject computer's case and the time constraints I had in getting this project finished, I limited this case mod project to being strictly externally cosmetic. Using your imagination on your own machine, you can easily go beyond what I did here.

So, grab yer scalpel, don your safety goggles and come up to the lab....and see what's on the slab........

SAFETY FIRST, AND ALWAYS

Before we begin, most, if not all of the construction and painting techniques I am going to discuss involve sharp knives, potentially dangerous power tools, paints and adhesives with noxious fumes, and many other ways to hurt yourself. I can not stress enough the importance of safety. ALWAYS wear eye protection when using power tools, especially when using them on plastic like we will here. Tiny bits of flying heated plastic and your eyeballs, do not a good match make. Also, ALWAYS work in a well lighted, well ventilated environment. The glues and paints used have nasty smelling fumes that love to wreak havoc with the human central nervous system. No need for duct tape and plastic here, just some basic common safety sense.

If you get careless and get plastic bits stuck in your eye, or super glue your fingers together (or worse: to your project), don't say I didn't warn ya.

GATHERING THE MATERIALS

Besides the old IBM PC300 PL with it's 200Mhz MMX processor, 40MB of RAM and 4GB hard drive, there is a long list of items that I used in this project. Most of which I already had lying around the house. Any well stocked hobby shop will carry 90% of the materials I used. The other 10% can be found in places like Home Depot (and I know *NONE* of you *EVER* go there). There is a list of suppliers at the end of this article should you need to track down any of the materials mentioned. Some of the materials I used are:

• Lots of spare model parts: I have several large boxes of spare parts I've accumulated in over 27 years of model building. Don't fret if you don't have as many as in Fig. 2. Any old bits of packaging or other stuff can be used: fast food drink lids, plastic milk bottle and detergent tops, old electronic components....you get the idea.

• **Strip and sheet styrene plastic:** Made of the same type of plastic that model kits are built from and available in just about any size and shape you can think of. Any well stocked hobby shop will carry this stuff.

• **Corrugated plastic wire housing:** Available at places like Radio Shack (I got mine at a dollar store), it's supposed to be used to channel your miles of computer cables into a neat, single conduit. We'll use it for something a little different here.

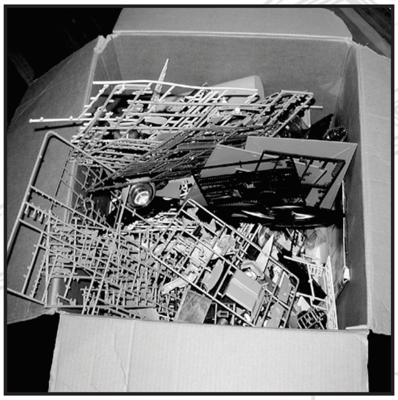


Fig. 2: Box of spare model parts.

• **"Eggcrate" drop ceiling-type light fixture:** I used this to make a monitor rest for the top of the computer. You can get this stuff at Home Depot and adds a great "industrial" look to any project when you paint it silver.

• **Spare model decals:** These are the "water slide" type that come with a model kit. I have a large collection of spares and some that I made myself, but any kind of decals, stickers or markings will work too.

• **Model paints (acrylic)**

• **Spray paints (enamel):** Spray paints made specifically for models are the best thing to use. Don't just use any spray paint from the hardware store, as it may react with the plastic of your case and turn it to mush. Check the can first to see what kinds of plastic it's compatible with. Krylon makes some excellent plastic-friendly spray paints, but model paints (like those made by Testors) are still your best bet. Also, do not apply enamel paints over acrylic paints. The enamel will react with the acrylic under it, bubble up and ruin the finish. It's OK to apply acrylics over enamels, however.

Also used were:
- 5 minute epoxy
- Cyanoacrylate cement (also referred to as "super glue")
- Masking Tape
- Dremel tool with grinding, drilling and sanding bits
- An airbrush with compressor

PREPPING THE PATIENT

I started by disassembling the case of the computer. Much like a model kit, it was assembled from several different subsections that were put together to make the whole thing. The front subassembly was connected to the main part of the case by four small screws, and contained the bezel housings for the CD-ROM and floppy drives, volume controls, and headphone jacks. Each of these components in turn had their own parts that were held in place by simple tab snaps that whenpressed inward, released the parts and allowed them to be removed (Fig. 3).

Since the computer case could be taken apart so easily, it made me think that the various subsections could be finished using different methods. I settled on two basic styles: a black and silver industrial/tech look, and what I call

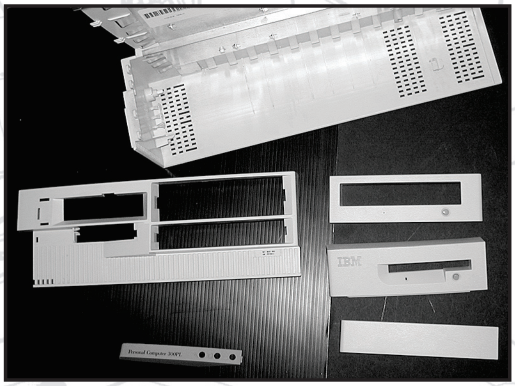


Fig. 3: The disassembled computer.

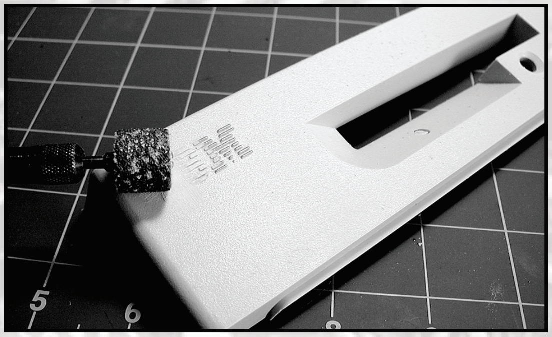


Fig. 4: Grinding off the IBM logo.

the "Borg assimilation/Weyland-Yutani atmosphere processing plant" look. Having two different styles of finish would make the project that much more visually interesting. For now, I'll focus on one component; the floppy drive bezel housing, as the construction techniques was similar for the other sections.

THE TRANSMUTATION BEGINS

One thing I wanted to do right off the bat, was get rid of the IBM logo on the floppy drive bezel. After a few seconds with a sanding drum bit on my Dremel tool, it was bye-bye to Big Blue (Fig. 4). With the Dremel, you have to use caution, not only with the barrage of flying plastic bits that will be looking to land in your eyes (you DO have safety glasses on, right?), but also with the speed and pressure applied to the plastic. The heat from the friction of the tool on the plastic will cause it to melt and distort if you're not careful.

The Dremel leaves the areas it sands kind of rough, so you'll want to sand it smoother before continuing. Your best results will come from using a range of waterproof sandpaper grains between 320 and 1500. Always sand with wet sandpaper, as it keeps the level of dust to a minimum, and gives a better finish.

OK, so we've sanded away the IBM logo. Now what? Time to start adding some detail parts. On each section I used a combination of random model parts from my spares box, bit of sheet and strip styrene, and some cast resin parts I had made a few years ago that were left over from other projects. In adding detail parts to projects like this, I try to follow a philosophy I like to call "structured randomness"; you want things to look like they serve a purpose, yet you don't want them looking to rigid or planned. You're going for more of a cluttered look than anything else.

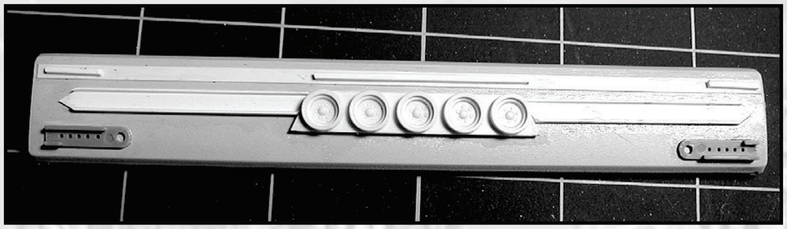


Fig. 5: The CD-ROM tray front cover with details added.

Depending on the type of plastic your subject case is molded in will determine what kind of adhesive you use to glue your parts on with. Luckily, the IBM case was made out of a plastic that responded well to the plastic cement that I normally use, something available at any hobby shop called **Tenax 7-R**. It's not so much a glue, as it is a "plastic welder". It melts the plastic at the points where two pieces join, just enough so that they bond together when the joint hardens. It takes literally just seconds to bond two pieces together, and they won't come apart easily. You DO NOT want to use the old fashioned tube-type "model airplane glue", as that stuff in a word, sux0rs. 5-minute epoxy will work very well for different types of materials too (such as bonding metal or resin to plastic). Fig. 5 shows the front of the CD-ROM tray after it has it's share of details glued on with Tenax 7-R.

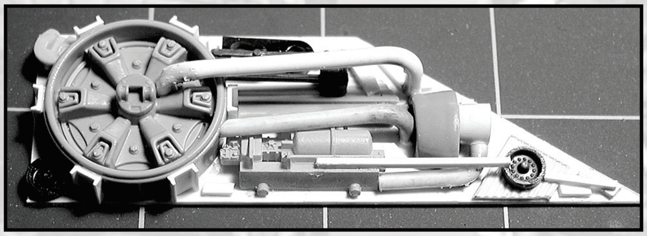


Fig. 6: The first detail panel takes shape.

Let's spend a minute on putting some spare model parts, bits of plastic and other odds & ends together to form a panel that will go onto the front of the computer. I started with what I think was a wheel hub from a tractor trailer kit. I really don't know where it came from, except that it was in my spares box for about 10 years. I added some bits of "U-channel" strip styrene around the diameter to try and disguise it a little bit. I glued this onto a oddly shaped scrap of sheet styrene, and started filling it up with other spare parts and bits of plastic (Fig. 6). When I was happy with the level of clutter I had achieved it was time to paint this panel.

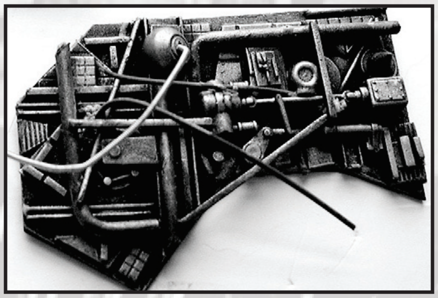


Fig. 7: A painted and drybrushed detail panel.

Everything at this point was white and light grey, as that was the color plastic everything was molded in. I wanted this to look like something that was made of metal, but was old, worn and used. Your first thought might be to start by painting it silver. Wrong answer! Just like the Rolling Stones sang, paint it black.

With a can of black spray paint, I gave the detail panel a thorough covering. Don't spray too close or too thick. Several light passes are better than one thick one, as you'll loose the finer details and it will take forever to dry. After the black dries, it's time to turn the plastic into metal. You'll want to use some metallic acrylic model paint for this. It doesn't matter what type of metallic color you use as long as it's acrylic; I used one called Graphite, which has a silvery-gunmetal look. Shake the bottle so you end up with some paint in the lid of the bottle. With a flat, wide model paint brush, dip the brush into the paint on the lid. Next take a piece of cardboard and rub most (but not all) of the paint off the brush. That's right. Wipe the paint off. Stay with me, were gonna make magic here. Now, take your brush and with short but fast strokes drag the edges of the bristles over the black areas of the detail panel. See what happens? The paint on the brush sticks to the highlight and raised areas of the parts on the panel. By using more pressure and repeating the paint on/paint off the brush procedure, you'll gradually built up the metallic areas, but the black will still show through and make it look rough and worn. This technique is called "drybrushing" and is great for picking out details and adding depth. With practice, you'll get some very cool effects. Fig. 7 shows a panel that has been drybrushed and finished. See how the black peeks through the metallic highlights? On this panel, I added some rubber tubing and bits of old guitar strings to simulate wiring. With the procedure I just described, you can go on and work on other sections of your case.

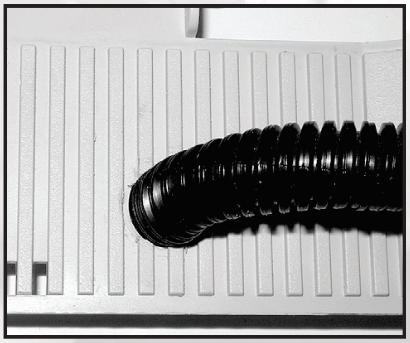


Fig. 8: Test fitting the conduit.

THE BEAST TAKES SHAPE

With the basic detail on the subsections underway, I next started on the main part of the case itself. It was mostly covered with a molded vertical grating. I decided to paint this area with the same silver/black drybrushing technique I described above. First, I had to prep it for a mod to the front of it. I drilled a large hole the same diameter as the flexible wiring conduit I had (Fig. 8) and test fit the conduit to make sure it would stay in place without having to be glued. The conduit is made of a funky, slick plastic that no adhesive will ever stick to, so it had to be a force fit. For the other end of the conduit, I made sort of a "junction box" from an old HO scale model railroad freight load (a cushion coil housing, to be exact). Fig. 9 shows how I added some random parts and plastic bit to it, and test fitted the other end of the flexible conduit. I sprayed this with chrome silver, and drybrushed it with black, so it would contrast with the area it would be glued to.

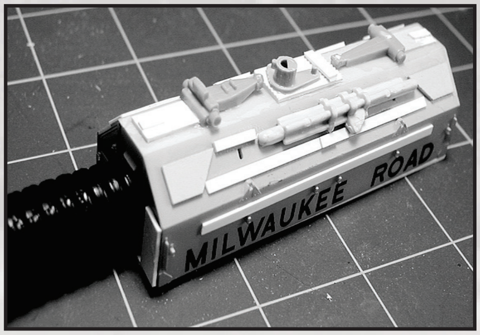


Fig. 9: The unpainted "junction box".

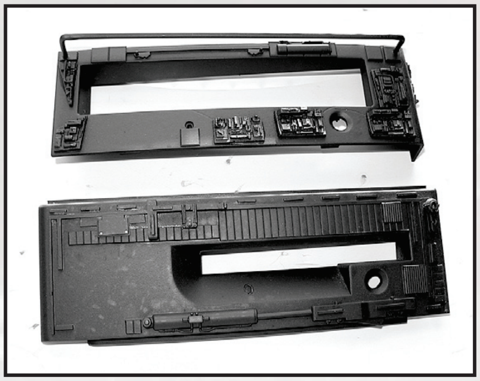


Fig. 10: Detailed and primed sections.

At this point, all the subsections had their detail parts applied (Fig. 10). I sprayed all the subsection with a grey primer coat and let them dry for a couple of days so the paint cured completely. While everything else dried, I sprayed the ridged area of the main case section black. Then I drybrushed it silver when it dried, just like we previously discussed. When this all had dried, I masked off this newly painted area to prep the case for the rest of the painting. I wanted the large, flat area on the top of the case, and along the right edge to have the worn, dirty look of plates or panels with different shades of color. With careful shading and a couple of special techniques, this look can be obtained with eye-popping results.

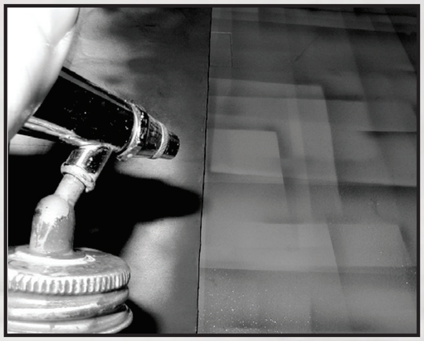


Fig. 11: Airbrushing along the masking.

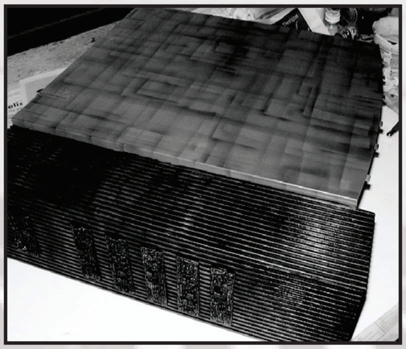


Fig. 12: Painting the main body is finished.

With the ridged areas that I painted black and silver masked off with tape and newspaper, I started by spraying a base coat of blue-gray along the top and side of the large, main case body. I picked out three different shades of grey, some black and a dark green from my acrylic model paint collection. This is where I switched from using spray cans to using an airbrush. Spray cans are great for covering large areas easily, but for close-in detail like we're about to tackle, nothing beats an airbrush. With some extra effort and careful

Fig. 13: Jazz things up with markings.

spraying you can get results similar to what I'm about to discuss, but you won't be able to get the same amount of precision and control that an airbrush provides. I also prefer to use flat (matte, non-glossy) paints as they tend to spray better and dry much faster than gloss paints do.

Starting with a shade of medium grey, I took a piece of plain white paper, folded it in half and laid it on the surface of the case. I started by lightly spraying along the edge of the paper, just misting the exposed area of the case. I kept the brush about 7 inches from the surface so the paint wouldn't go on in too concentrated an area, but rather faded into the base coat as it got further away from the edge of the paper (Fig. 11). When I removed the paper, I had a nice, crisp demarcation line that faded into the base coat. I repeated this by rotating the paper, spraying along different corner, but all the time keeping everything at right angles. I rotated through the 4 or 5 colors I had selected, saving the black for last. When I was finished, I had a random patchwork effect that had a layered and feathered appearance, thanks to variety of shades and colors I used. This process was used on all but the black and silver components of the case subsections. Fig. 12 shows the main body section after this process was finished, and the masking removed from the rest of the section. With all sections now painted, lets move on.

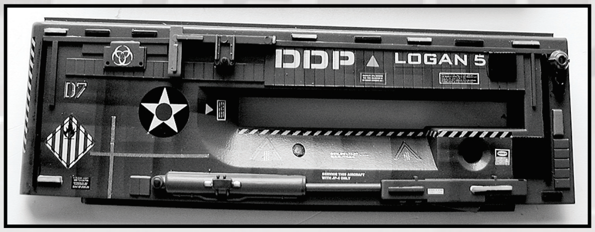


Fig. 14: The floppy drive bezel with decals added.

MARKINGS AND LETTERING

Even though you might be tempted to paint your case and call it done, lets go one step further and add some markings and decorations. Since we're following SOP for model building with this project, I used the "water slide" type decals that come with just about any model kit that you buy. I have MANY spare decals and even some unused complete sets, as well as a few sets that I printed myself for other projects (Fig. 13). But you can use just about anything that will adhere itself to your case (like that anti-RIAA sticker you've been trying to find a good home for). The advantage of model kits decals, is that they are printed on a very thin, flexible, clear material that can be made to snuggle down over tiny details so the decal looks as if it painted on.

Before applying water slide (model kit type) decals, you'll want to have a nice, glossy finish on the model. Applying decals to a flat, dull finish traps tiny air bubbles under the decal that turn silver when the decal dries. Having a smooth, glossy finish will make the decals much happier and less prone to air bubbles. I sprayed all the components with several light dustings of Krylon Clear Acrylic. Again, it's better to spray several thin coats, rather than one thick coat. Once the clear coat has dried for a day or so, start slappin' on some decals (Fig. 14). I used a decal solvent called Solvaset that softens the decal film and pulls it right over the tiniest of details and make it really look painted on. This is optional, but the decals will look much better. If you choose to try this, DON'T touch the decal for at least two hours so it can dry. You will ruin a decal if you touch it with wet solvent on it. Once all your decals are dry, you'll want to give the case one more clear coat. I used a flat clear Krylon Acrylic, but you can use a gloss if you want. This final clear coat will seal and protect the decals, and help blend them into the paint job. ***NOTE***: If you are not using water slide decals, but rather the standard peel-n-stick type, there is no need to apply an initial gloss coat. However, I highly recommend you apply a clear coat anyway, as it will protect your paint job from scratches and other damage.

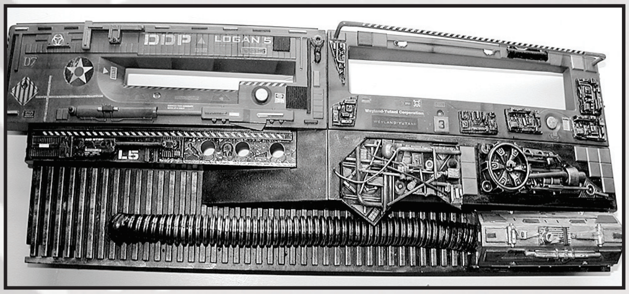


Fig. 15: The finished sections are reassembled.

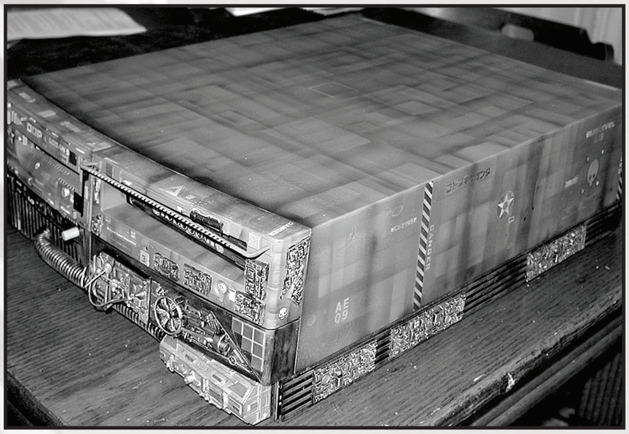


Fig. 16: The main sections are put back together.

PUTTIN' IT BACK TOGETHER

With all the dirty work done, all that's left is to reassemble the case and put it back on the computer. You'll want to use some extra care, as not to break off or damage any of the

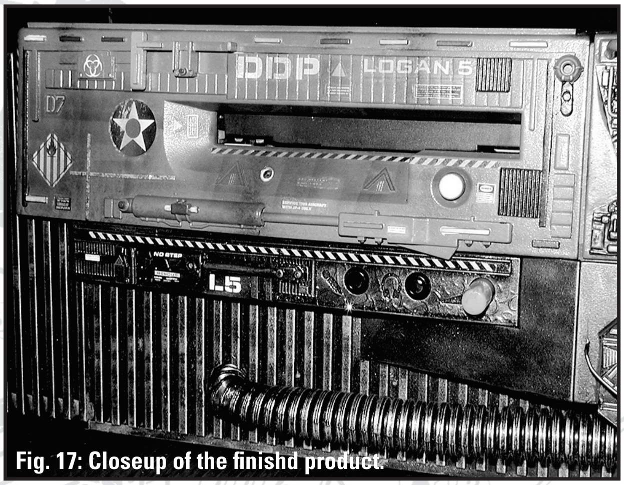


Fig. 17: Closeup of the finishd product.

details parts or your cool ass new paint job. Once you have the case put back onto the computer, you can finally see how all the parts and paint work together, without a hint of beige remaining (Fig. 15, 16, 17, 20, and 21). Since this is a desktop machine, the monitor is best placed on top of the computer. But since I spent all that time on painting the top of the case, I didn't want some damn monitor scuffing it up. So, I took a hunk o' "Eggcrate" style drop ceiling light fixture, cut it to fit on top of the computer, and sprayed it chrome silver. This monitor stand helps protect the paint on the case, and the paint job is still visible through the openings in the Eggcrate. In Fig. 18 you can see the Eggcrate, as well as the open CD tray with it's newly modified bezel.

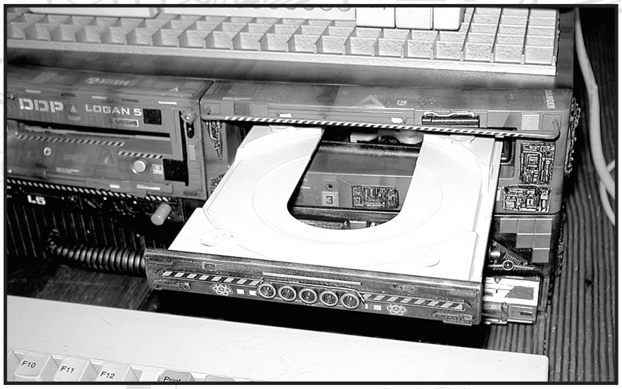


Fig. 18: The CD-ROM tray and Eggcrate monitor stand

Dammit, Jim! It's a computer, not a model!

Not content with just sticking this thing on a table and letting it collect dust, I decided it would be fitting to install Linux on this newly resurrected IBM PC. When I hauled it from the dumpster, it had Windows NT 4 installed. Sorry, I don't do Windows. But with only 40MB of RAM (and additional RAM just about impossible to find) I couldn't install just anything. Using some helpful info gleaned from a great article written by dual_parallel (posted at ***www.oldskoolphreak.com***) about running fat Linux distros on thin hardware, I installed Mandrake 7.2 with IceWM as the window manager (Fig. 19). It's not a rocket, but it runs really well and is a great machine to learn and explore Linux with. And it was free.

I first installed Red Hat 7.3, but it wouldn't recognize the NIC. Mandrake works like a champ. As you can see from the Fig. 19, the monitor is still beige. A continuation of this project would be to mod the monitor to match the computer. This could be done my making more detail panels like the one shown in Fig. 7.

Hopefully I've motivated you to try this physical form of hacking and come up with your own case modeling project. I've posted color pictures of this project on line at: ***homepage.mac.com/loganfive/mod***. Seeing the mod in color might help give you some additional ideas. OK. Class dismissed. Go forth and make cool stuff!

Suppliers: Any well stocked hobby shop will have the items I used. If not, any paints, supplies and tools can be ordered from Wm. K. Walthers. Most hobby shops sell their catalog. If you can find one, order it direct: ***http://www.walthers.com/*** 0101

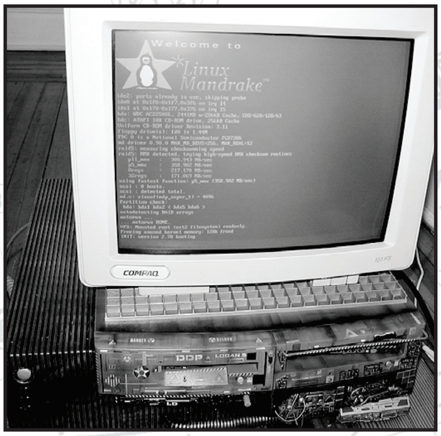


Fig. 19: Booting Mandrake for the first time.

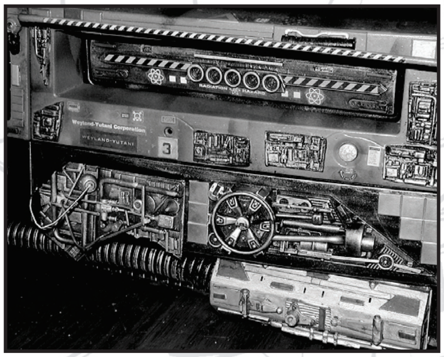


Fig. 20: Closeup of the CD-ROM docking bay.

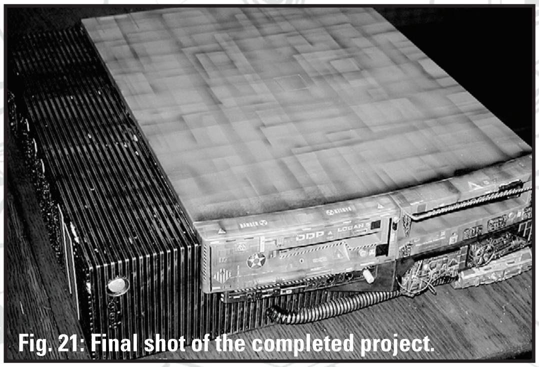


Fig. 21: Final shot of the completed project.

Big honkin' shouts to everyone in DDP. You guys rawk! Big honkin' knarly shouts to dual and StankDawg for getting me motivated to sit down and write something.

TWEAKING T-MOBILE

BY GADGET

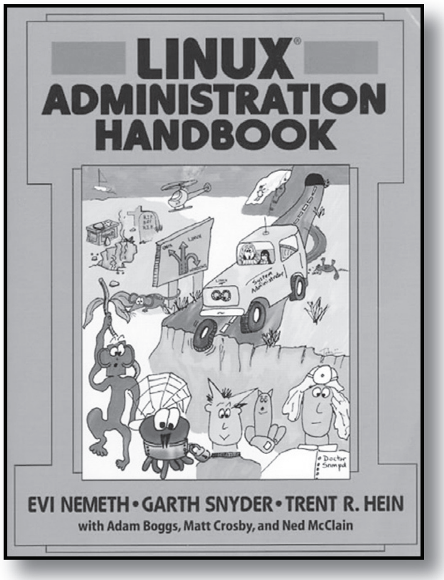
IT seems that we hackers are constantly looking for the newest coolest tool for our gear bags. I definitely fall into this category, but often find that my budget doesn't allow me to accessorize to my liking. Like many of you I have seen the new Pocket PC phones and couldn't wait to get one until the $500 price tag ruined my day. So what is one to do you ask.

Well for me the answer was to look into the past. I use T-Mobile phone service which is a GSM provider here in the US and abroad. The nice thing about this is that I can slip my sim card into any unlocked 1800 Mhz phone and immediately use it without having to call anyone to switch my ESN or other data. During the year 1996 or 1997 T-mobile was known as powertel and one of their phones was called the Nokia 9000iL. This phone was unique for the time because it was also a PDA with a clamshell design running the GEOS OS. Well I searched E-bay and found a guy who has several hundred of them still in the box. Needless to say I made a buy it now purchase and had the phone in two days all for $73 dollars. Thats about a $427 dollar savings over the pocket pc phone. So what you say why would anyone want some piece of junk phone from 1996.

Well the list of features speak for themselves. The phone has a greyscale screen with backlight. There is an HTML browser installed and you can access POP and IMAP mail. Fax service is avaliable if you subscribe to it and there is a built in fax viewer for viewing received faxes. SMS service is also included. In addition to the online features the phone has an address book, notes, and calendar. There is no GPRS capability for this phone so all online features are accessed through my monthly alloted minutes (3000 anytime). This works out pretty good since I dial the internet through my local ISP and can pull down my POP mail. I will admit that the speed is pretty slow (9600 baud), but that is ok when I can get this service anywhere I have coverage. To use the fax service I called T-mobile and spoke to the wireless data group and had a service called CSD activated for $9.99 per month. This is a neat service that provided me with two new phone numbers for my phone, one for faxes and one for data. Again all the fax and data calls count against my bucket of monthly minutes. One thing I like about my setup is that I can be on a voice call and receive a fax at the same time. I have had this service for one month now and am quite satisfied with it. The only drawback so far is that my battery only lasts about 10 hours in my phone so I have to carry a spare.

I forgot to mention the phone also has telnet. For those of you like me don't forget to look to the past and you might be surprised at what is available at a reasonable cost. 0101

man/pages/*.*


LINUX®
ADMINISTRATION
HANDBOOK
EVI NEMETH • GARTH SNYDER • TRENT R. HEIN
with Adam Boggs, Matt Crosby, and Ned McClain

Linux Administration Handbook: by Evi Nemeth, Garth Snyder, and Trent R. Hein.

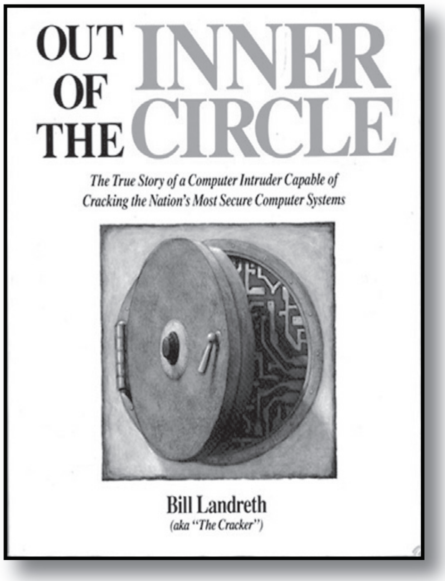
review by: coleco

UNLESS you are a fan of reading textbooks I can't imagine anyone actually sitting down and reading this book. Don't get me wrong, the book contains loads of useful information but is not something most people would readfrom end to end. The book was required for a Linux class that I was enrolled in but I preferred sitting in front of the terminal window rather than reading chapters out of the book.

The book is designed as an aid to becoming a Linux System Administrator. There are 29 chapters divided into three sections: **Basic Administration,** (file systems, serial devices, backups) **Networking**, (TCP/IP, routing, Domain Name System) and **Bunch O' Stuff**, (printing, performance analysis, daemons.) I am sure the authors thought they presented the information in a logical progressive manner but the chapters are not necessarily consecutive. One can skip from chapter to chapter and learn information as needed. The chapters are concise and concentrate the reader on learning particular commands or specific topics

The book covers three distros in detail, Red Hat, SuSE, and Debian. One point the authors make again and again is that Linux is a derivative of Unix and they often refer to various Unix methods and commands. The book does a good job at presenting commands and their multitude of flags on clear charts. The book also gives good descriptions of what information is contained in certain files. This helps the reader understand what reference the files contain and what the user may be changing. The book gives examples of how the original file should appear and how it should appear after following changes they recommend.

I found most of the information accurate and up to date and easy to understand. I wish the book had an appendix of the most popular commands and all their flags. I frequently found myself searching the index and flipping to various pages to find certain bits of information. Nonetheless, the Linux Administration Handbook is a decent reference book to have whether you are just starting out with Linux or have been using it for years. I don't know how many Linux handbooks you need in your library, but I would only consider this a supplement to others.


OUT OF THE INNER CIRCLE
The True Story of a Computer Intruder Capable of Cracking the Nation's Most Secure Computer Systems
Bill Landreth
(aka "The Cracker")

Out of The Inner Circle: by Bill "The Cracker" Landreth.

review by: *coleco*

IF you are interested in quality "old skool" books, look no further. Written twenty years ago this book could very well have been the inspiration for Kevin Mitnick's ***The Art of Deception***. Although the computers mentioned in the text belong in the computer museum the lessons to be learned still apply today: secure passwords, educating users, hacking techniques. It is all there and all relevant.

The book talks about the exploits of a teenage hacker in the early 1980's. It mentions The Cracker's first exploits on a DEC-20 and TRS-80 and talks about his rise to fame among BBS. He started not knowing anything about computers but soon was so hooked that he read anything he could find on the subject, asked many questions, and spent long hours on his computer. In 1982, The Cracker and few fellow hacking friends started The Inner Circle and the book details some of their adventures and exploits.

The book was written with a wide audience in mind, so it not overly technical or dry. The reading is easy and tries to explain to the non-hacker how a hacker thinks and hacks. Out of the Inner Circle gives an interesting perspective on the types of hackers: "novices, student, tourist, crasher, and thief" and to what level they hack. The book gives a brief history of hacking but concentrates mainly on underground culture of the time: Captian Crunch and ***Wargames***. It describes Trojan Horses, Trapdoors, Logic Bombs and the like and then talks about security.

The second half of the book is all about security. How much to worry about it, how to discourage hackers, how to secure your system. It amazes me how familiar Bill Landreth's advice is and yet twenty years later people still do not follow it. The book talks about how to secure the technology of the time and ends with how The Inner Circle got taken down by the Feds. The appendix is a great stroll down technology lane, discussing the security devices of the time: callback devices such as Sleuth and Data Sentry, the Sherlock encryption device, and others.

Although the book is dated it is an enjoyable read. Not only did it bring me back to my hacking youth but it also presented topics that are still relevant today. Although it might be hard to find, it is a good weekend read.

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.

A Physical Security Primer for the Community

by dual_parallel
www.oldskoolphreak.com

"There are three simple principles to follow: keep people away, keep them out, and protect your plumbing."
Microsoft 5-Minute Security Advisor - Basic Physical Security

Unfortunately, physical security is not as simple as most IT-centric books and websites lead you to believe. Effective physical security is not a checklist. Effective physical security comes form a methodology that is applied system-wide (the idea of "system" discussed later). This article will present the foundation of a methodology that can be used to secure assets of any value, and then discuss a few specific technologies and how they are used (and misused) in the methodology.

Checklist? Methodology? Even more basic: What is physical security? First, what physical security is not. Physical security is not safety. For example, protecting your lab from accidental fire or storms and keeping harm from humans from said fire or storms is not security; that is safety. Security is the protection of assets from humans with malevolent intent, hereon called adversaries. Assets can be material objects, information or human life.

It was said that a system-wide view to physical security must be taken. Here, system does not mean a single computer or network, but the entire entity within which an asset resides. To the example of a hacker's lab, the computers within the room, within the house, the people that reside and have access, any security technology used, and any off-site security elements. That is a system. Every element interacts in some way.

Methodology

With security and system defined, it is time to describe the methodology. As stated, the object of physical security is to protect assets from adversaries. So defining what is to be protected and its value is the first step.

Let's take a hacker's computer containing dairy-fresh exploit code (ready for full disclosure). To the hacker, this is a high consequence asset, meaning that loss of the asset is unacceptable. An off-site backup would mitigate risk, but let's assume this is not possible. In this system, the greatest amount of resources will be spent securing this asset. Suppose this hacker is also an avid gamer and must protect a rare collection of 2D Saturn shooters. The games are important, but not as important as the noteriety from such a fantastic piece of code. Therefore, less resources will be allocated to protecting the Saturn discs. As can be plainly seen, defining what to protect and at what level is a first and important step.

As disturbing as this scenario may be, the hacker knows that a younger sibling covets the rare, round pieces of plastic. The desire for a classic, Japanese, vertically scrolling shooter transcends the notion of family - the young one is the adversary.

Knowing the adversary, the hacker simply hides the discs or locks them up, easily foiling the sibling's plans. The key here is that the hacker knew of the threat. Determing the adversary is the next part of the methodology.

Along with knowing what assets are critical to protect, knowing what adversary to protect against will determine the resources that must be allocated to establish effective security. With an adversary that's in grade school, little to no resources are needed to protect the Saturn discs. If the rogue nation state of Japan wanted to liberate the rare games using espionage, the hacker might not deem the discs worthwhile to protect due to the resources needed to protect them from such a threat. More likely, the hacker would not have the resources to protect against such a threat and would have to deem the scenario as acceptable risk.

A more credible scenario would include a script kiddie neighbor who has seen some time in juvinile detention. A find like the hacker's code would be worth a simple B&E. This scenario will require much more resources than the familial adversary - the adversary has greater capabilities and motivation (or lack thereof) and the asset is much higher consequence. With those two things determined, the hacker can begin to add effective security to the system. What should the hacker do?

Simple. The hacker just installs six CCTV cameras around his house and therefore has effective security.

Wrong. Throwing technology (especially cameras) blindly at a problem is never the solution. Effective physical security consits of three elements detection, delay and response. Each will be explained, in order, to show how the methodology provides effective security.

Detection is the first peice and for good reason. If no one is home when the kiddie launches his (physical) attack, and there is no security system, the kiddie will go undetected and have as much time as he needs to get the code he wants, whether by accessing the compluter or stealing it outright. Without detection, the adversary has the time to complete necessary tasks unhindered and walk away with the asset. with detection, let's say a passive infrared sensor (PIR) or a balanced magnetic switch (BMS), both sensors that provide detection, the adversary must complete necessary tasks (defeat delay) before response arrives. (There's some bad news for the hacker when we reach Response.)

Delay, the next piece of effective security, must occur after detection. Delay is the implementation of technology or procedure that slows adversary progress. In the sibling scenario, technological delay (a locked box in a closet) could be implemented, sufficiently delaying the adversary (with a given set of capabilities) to allow response (the hacker responding to a door alarm) before the adversary completes his tasks. Procedural delay (hiding the discs) may achieve the same goal.

To review, if deteciton is placed after delay in the adversary task timeline, or there is no deteciton, the adversary can defeat any delay and achieve his goal. To protect an asset, there must be detection to know that there is an attack, and there must be delay after said detection to allow response.

Response, simply put, is the good guys catching the bad guys - the police responding to a burglar alarm. The bad news for the hacker (and most homeowners, unfortunately) is that local police response time is usually much longer than the average burglar's task time. The script kiddie can smash a window and walk out with a tower much quicker than the police can respond. That's where our friend delay comes in - slow the adversary to where he cannot complete his tasks before response arrives.

So you can see that effective security must have detection, delay and response, in that order. The value of the asset will determine the amount of effort and resources you allocate to secure it. But what technologies are effective and where do they fit within the methodology, you ask? Let's discuss a few that are pertinent to a hacker.

Cameras

Cameras are misused so much that they deserve their own section. And to sum up their misuse, only one statement needs to be made: Cameras are not sensors.

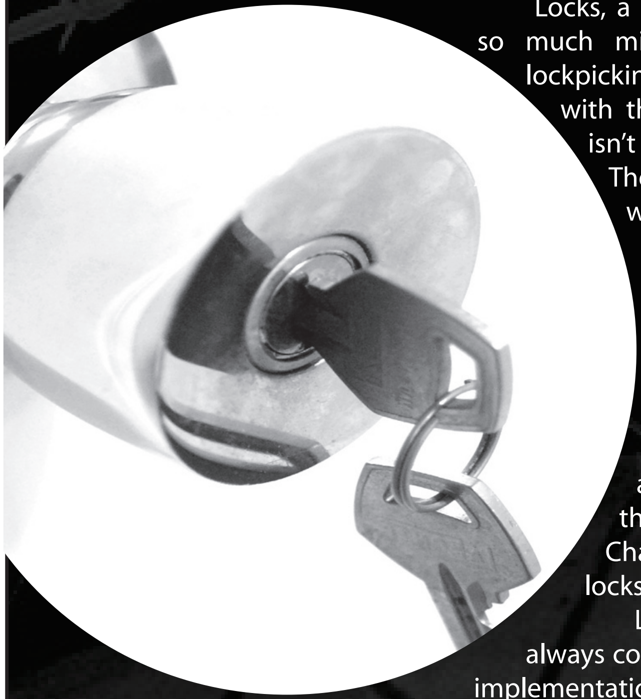
Cameras in the above methodology would be used as assessment after detection is made. (Of course, this is a simplified generalization sufficient for this introductory paper.) Adding 20 cameras to a security system, and expecting a human to watch the associated monitors to detect wrongdoing, is a detriment to security effectiveness. The human is the detector, and a poor one at that. Humans are good for about two hours, are prone to distraction, need to eat, go to the bathroom, etc.

Again, don't rely on a camera for detection. Get a good sensor, with a good anunciator (ideally, covert to the adverasry), and use the camera to assess the situation.

Cryptography

If you are familiar with distributed.net, then you know that it takes essentially immense resources to crack common encryption - the equivalent of 160,000 PII 266MHz PCs crunching approximately 1,700 days to find a 64-bit key. All but governement entities have such computing power, or motivation. So it can safely be assumed that encryption is effective delay. And delay it is.

Data is a funny asset. Once it is "0wn3d" it loses its value: infinite copies can be made and can never be taken back. So, if encrypted data is unknowingly (to the owner) intercepted, or to tie this subject to the physical world, the entire box stolen, there is no detection and the adversary has unlimited time to defeat the delay (the encryption). But good encryption is extremely effective delay, ensuring the retention of the asset's value.



Locks

Locks, a fascinating technology to be sure, are not so much misused as they are misunderstood. Take lockpicking for instance. Anyone familiar in the least with the sport of lockpicking realize that a thief isn't going to pick the lock on your back door. The thief is going to throw a brick through the window and enter that way. Locks are delay with respect to the portal they are installed. A Sargent & Greenleaf lock on a hollow laminated door would be foolish, as would a cheap Master combo lock on a steel-reinforced concrete door.

Take this and consider computer tower or laptop locks. They offer delay, but little delay at best. And this delay does no good when the entire computer is stolen (screwdriver and Channellocks, anyone?). Even the laptop cable locks can be defeated with a simple hammer.

Locks are a good idea, if not essential. But always consider adversary circumvention and the true implementation of locks as delay.

Take this methodology and the technologies discussed and use the information while observing security technologies and procedures everywhere you go. Look around every building you enter- Are there glass break sensors? Security cameras? Security guards? - and apply the methodology inside your head. You'll begin to understand the strengths and weaknesses of (basic) security systems. And hopefully you'll start to consider, and possibly research, more advanced technologies (biometrics, x-ray, microwave sensors, etc.) that you may encounter.

Kismet ON KNOPPIX HD INSTALL

By: bland_inquisitor (bland@binrev.com)

I spent the better part of a weekend getting Kismet to work on my Knoppix HD install. I thought I'd write this up so that you can get the thing to work in a matter of minutes. I am using Knoppix on my Dell Inspiron 8200 with an Orinoco silver 802.11 card.

What you will need:

- Knoppix 3.2-2003-06-06 installed on your hard drive
- Orinoco driver patch
- About 20 minutes

There is a problem with the Knoppix hard drive install script on most versions prior to the one listed above. Save yourself much gnashing of teeth and be sure to use the latest Knoppix distro. The version of Knoppix we are dealing with does not use the latest version of Kismet, so you'll need to get it like so:

```
[user@lappy]$  su –
[user@lappy]$  password
[root@lappy]#  apt-get update
     *much stuff going on*
[root@lappy]#   apt-get upgrade
     *much stuff going on*
[root@lappy]#   apt-get install kismet
```

You should get a prompt telling you that Kismet is already the latest version, this is good. Next, you'll need to alter the Kismet.conf file to suit your needs. I had a problem getting Kismet to work using the default log template, so I recommend you change line 203 to something like:

```
Logtemplate=/home/Knoppix/Desktop/Kismet logs/%n-%d-%i.%l
```

Remember, there's a period after the **%i.**

Next, you will need to upgrade your Orinoco firmware at ***airsnort.shmoo.com***.

```
[root@lappy]#  cd /lib/modules/2.4.20-xfs/kernel/drivers/net/wireless
[root@lappy]#  wget http://airsnort.shmoo.com/orinoco-0.13b-patched.diff
[root@lappy]#  patch –p0 < orinoco-0.13b-patched.diff
```

Then when that's finished, run the following:

```
[root@lappy]#  iwpriv eth1
```

Look for an entry called monitor, if you see it (and you should) then you're in business!

There is an optional step, spoofing your MAC address. Since Kismet is a passive scanner, there is really no way for someone to pick you up wardriving, but if you're über-paranoid, there are two ways to do it: doing it by hand, or using a t-rad perl script.
To manually do it you need to do the following:

```
[root@lappy]#  ifdown eth1
[root@lappy]#  ifconfig eth1 hw ether 00:00:00:00:00:00 (or whatever MAC you want)
[root@lappy]#  ifup eth1
```

Then run ifconfig to make sure that your new MAC is in place.

The second way is to download dual_parallel's "macninja" script at ***www.oldskoolphreak.com*** and run it to grant yourself a random MAC address.

All that's left to do is to test Kismet on your box.

```
[root@lappy]#  kismet_monitor –H
       *output*
[root@lappy]#  kismet
```

Here you should be at the pretty Kismet GUI, and your WAP should be in view. All that's left to do is go wardriving and have phun.

After your Sunday wardriving, you will need to take your card out of monitor mode to be able to use your internet as normal.

```
[root@lappy]#  kismet_unmonitor
     *output*
[root@lappy]#  /etc/init.d/pcmcia restart
```

And bada bing, you're good to go.

I hope this has helped you to negotiate the trials of making Kismet work. And remember bland's #1 rule of hackerdom: "Don't be a dick"

Shouts: StankDawg, w1nt3rmut3, dual_parallel, Sean Kennedy, hacnslach, and everyone that supports *Binary Revolution Magazine*.

FUTURE OF TELEPHONY is VoIP

By: Epiphany

Anyone would notice, if they were browsing through the Request For Comments server, that many of the latest RFC's deal with revisions in the "Voice over IP" protocol as well as new theories suggesting how to best implement the technology. This in itself already suggests that this technology has potential. However, before anything we must understand just what "Voice over IP" is. VoIP, also called IP telephony, promises to unite and integrate the role of the hacker and the phreak for it is the concept of creating a telephone system similar to a PBX but run through the TCP/IP protocol of a company's network. The benefits of such of set up are great in number; one of the most important, and controversial, however is a company finally being able to cut the telecom industry out of the picture. For example companies running a VoIP network no longer need to depend on Telecom Certified Workers to administer and repair the old proprietary PBX as well as being able to set their own rates on calls within their network.

The basic structure of a VoIP network is not unlike a normal computer network. In fact the whole idea of this technology is to have the system run on top of an already existing computer network. Therefore all the devices and possible vulnerabilities that many of you are already familiar with still exist. VoIP systems can be created using two different protocols. One being the "H.323" which is based on the structure of the telecom SS7 (Signaling System 7) and the increasingly commonly used SIP (Session Initiation Protocol) which is based on the structure of HTTP. Despite their slight differences both protocols are intended to do the same thing, which is to successfully communicate with the PSTN or Public Switched Telephone Network. The PSTN just another name for the network that MaBell runs.

As stated before, VoIP uses devices that are very similar to TCP/IP devices. For example in the SIP protocol the device that acts as the DNS server, meaning resolves PSTN Numbers (telephone numbers) to IP addresses that can be used to find devices on a network, is called the Registrar/Location Server. Also, the so called web server of VoIP is called the SIP Redirect Proxy. One necessary device that cannot be compared to anything on a TCP/IP network is the Media/Signal Gateway whose purpose is to serve as the connection between a VoIP LAN and the PSTN. The Media/Signal Gateway has several voice ports which connect to voice trunks on the PSTN. The SIP protocol itself is an ASCII based protocol that acts as both client and server. The service typically binds itself to TCP port 5060 and uses various UDP ports for data transfer. SIP evokes yet another protocol to do its transporting dirty work. This protocol is called Real-Time (RTP). All of the above devices are necessary for the three types of functions that makes VoIP what it is:

VoIP to VoIP - Calls which are either calls within the network or calls routed through the internet to connect one VoIP network to another.

PSTN to VoIP - Calls which originate from the PSTN network and need to be connected to an existing VoIP network.

VoIP to PSTN - These are calls made from a VoIP network to the existing PSTN.

These three situations are all made possible through the packet structure of RTP. Although similar to TCP packets they are modified to include additional data vital to a VoIP network. (For more in detail structure of RTP packets check RFC file 1889)

Now it is time for the part that is important to us hackers and phreaks; known vulnerabilities in VoIP networks. One of most well known vulnerability was posted on ***www.securiteam.com*** on May 2002 in which it was discovered that the VoIP phones themselves were vulnerable. The Cisco 7900 line of VoIP phones included a built in web service running on port 80 which contained debug pages as well as status information. Not to the surprise of many this service has no authentication on the pages and contained exploitable scripting errors. It was also possible to DOS attack the phone by typing in this URL ***http://<phoneIP>/StreamingStatistics?33000***. This URL caused an error which successfully caused the phones to reset. Another URL which was far less destructive but good for footprinting is ***http://<phoneIP>/PortInformation?<numberofport>***. This link gave information on certain TCP ports on the phone. The final known vulnerability in this product line is that if someone had local access to this model phone they could access and change the settings on the phone by pressing the settings button and punching in the string "**#" on the phone keypad.

There are several other little known vulnerabilities in a VoIP network itself. These were suggested by Brennen Emerick Reynolds. The first is the VoIP version of a Ping, by sending a CANCEL request to a VoIP terminal it will reply with "Transaction Does Not Exist". It is possible to send this request through entire subnets of IPs and discover which ones are computers and which ones are VoIP terminals by their reply. Even though beige boxing is not possible because of the upgrade from RJ-11 to RJ-45, being on a network eavesdropping and sniffing is still possible. In fact there is already a utility available for decoding captured RTP streams. It is called VoMIT (Voice over Misconfigured Internet Telephones) and is available at ***http://vomit.xtdnet.nl.*** The final vulnerability, besides the lame SYN floods that are possible, is Connection Hijacking. Someone who is knowledgeable with the SIP protocol can route and redirect existing sessions by sending false response messages. For example by making sure certain sequence numbers and identifiers on an RTP stream are higher than a target's stream, it is certain that the target will unknowingly accept a stream from a fellow hacker/phreak.

As you can see, VoIP is truly a revolutionary technology. Already companies are providing service to internet users that is much cheaper than if it were run by the telecom industry. And of course the MaBell giants are trying to stop the spread of this open-source technology by trying to influence Congress to pass laws regulating VoIP (***http://pulver.com/reports/statesfightvoip.html***). Hopefully though the technology will spread despite their efforts so the common person can be able to learn and discover new things to do with VoIP networks. On a final note the only con to VoIP is that since it is a new technology many of the systems don't pass ANI correctly and I'm sure CallerID is exploitable.

Sources: RFC 2833, RFC 1889, RFC 2543, *Master's Thesis On Enabling Secure IP Telephony in Enterprise Networks (STEM)* by Brennen Emerick Reynolds.

BEST BUY Insecurities: REVISITED

By w1nt3rmut3
mut3@oldskoolphreak.com

Within days of the realease of my previous article, ***Best Buy Insecurities***, in the Spring 2003 edition of 2600, Best Buy revamped their security policies and even the employee interfaces. They have implemented changes in their network to defeat my previous "hacks". This article will focus on both those changes, and items of interest I didn't mention in the last article.

Items not mentioned

I couldn't stop finding tidbits about Best Buy and their network, so I continued my spelunking. I found that they have an on-floor "home page" if you will, called toolkit, that is link heaven. "toolkit," if you remember, wasn't accessible before, but I have obtained the keystroke for this. It is: **Ctrl + Shift + T, Ctrl + Shift + K**

This breaks you out of the demos and to the employee homepage, toolkit. This gives the user a very organized list of links to Best Buy sites such as Tagzone, MSI, Raincheck, et cetera. Originally, I had to go into the History of Internet Explorer to find my links. This method is much easier. My method of obtaining this will remain hidden, but you can SEE it very clearly. (Update: The company recently changed this. Now it is a different code, and the splash screen is protected by a login/password box. Some toolkits might work with Z Z or Z Z A instead of T K. No worries though, a simple surf should do the trick).

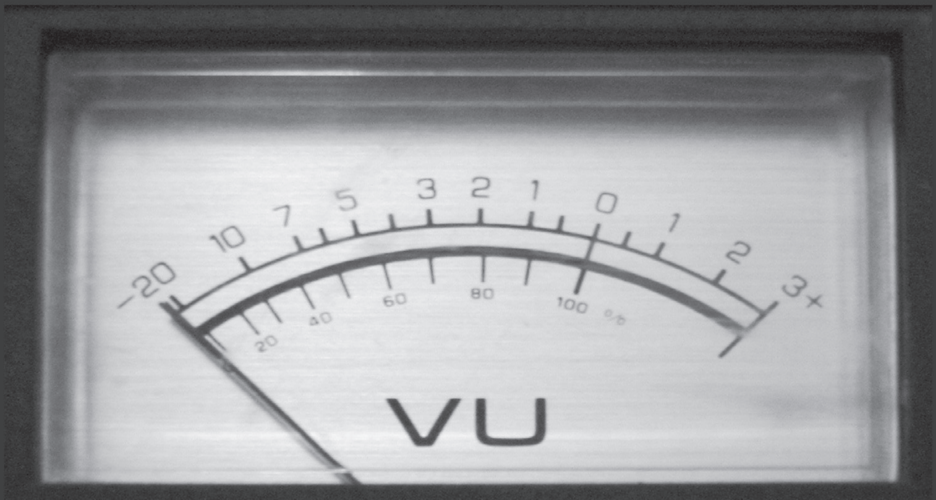
There is a multitude of standalone machines in each store, ripe with default settings. Now it is widely known that Best Buy, along with other stores, password protect screensavers and "interactive demos" so you can't get inside. Well the easiest way around that is the multimedia buttons the new fangeled keyboards have, but you probably knew that. Another way is the shoulder surf or brute force manuver. Yet another way is calling the store saying that you bought a floor model, and the screensaver won't let me in. They will give you the password. Something pretty cool that you might not of noticed is a random combonation of letters and numbers in the corner of the interactive demos. By clicking these, you immediately break out of the demo, and you got your desktop. The demos usually aren't live on the net, so they only get you so far. Still fun though. I suggest getting some type of recordable removable media and "borrowing" the interactive demo. Loads of fun.

Changes

The biggest would be an increased security and update policy on internet enabled machines. Even before the article went to print, machines I used at my store had patches on, along with stricter polices, such as no drive access. This might not be nationwide, so check with your stores (if you haven't already). As I mentioned before, you could only access certain sites, such as ***bestbuy.com*** and ***microsoft.com***. Further research reveals the fact that all of the IP addresses Microsoft owns (commonly referred to as a netblocks) are accessible. This could of been done for complete compatibility with any changes to the Microsoft Windows OS.

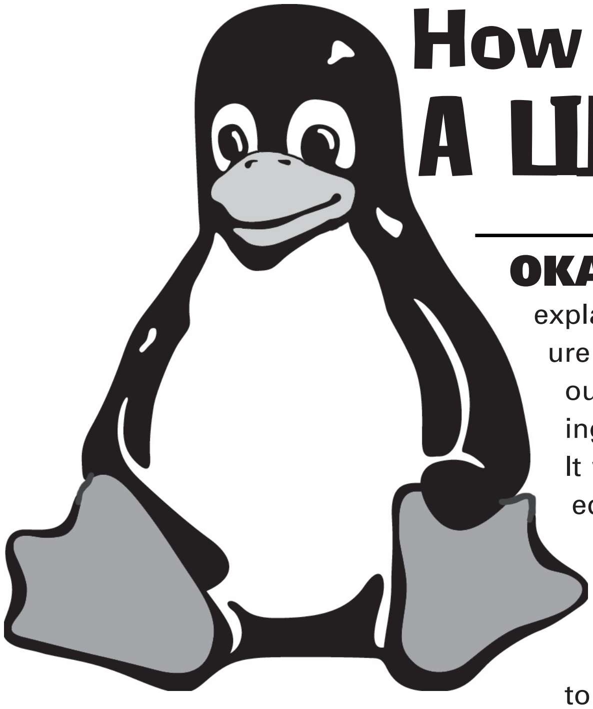
As previously stated, toolkit is locked down. They have moved to a new Version, 2.0. It's slicker and more up to date. Something else that I recently learned was that Tagzone is deprecated. It's popular brother, Retailzone, is used more. The login for toolkit is also different from the login to the store registers. The new login is best guessed to only begin with an A or J, and the six digit employee code. Retailzone itself needs a login, but no password. Shouldn't take too long to get by.

SHOUTS: All the peeps of Chicago 2600 because I left them out of the BBY Part One, the DDP crew, and to nomad and zack for their invaluable insight into the Best Buy network.



How To Configure A LINUX KERNEL

by BoBB



OKAY, in this file I am going to attempt to explain what exactly is required to configure a kernel from scratch. I'd like to point out straight from the beginning that learning to configure a kernel is no small task. It will take time and you will need knowledge of Linux. If you dont know what you are doing it can be very dangerous to your system to boot a custom kernel. I would like to spew out the standard CYA disclaimer JIC. You will need to be root to do most of the stuff in this howto, and everything done is tested on my setup and has not caused any problems. But every system is different and don't do anything you don't think you should do. I take NO responsibilty for what happens - this is just a guide. And now on to the good stuff!

First thing we want to do is get the latest stable kernel sources from ***http://kernel.org***. It will say on the front page what the latest version is, but at the time of writing it is 2.4.21. Okay, once you have the sources downloaded, cd to /usr/src and untar the kernel like so:

```
# tar -xjvf linux-2.4.21.tar.bz2
```

That should create a directory called linux-2.4.21 or something to that effect. You will need to make a symlink to this directory called linux so that other programs can locate the kernel sources when compiling. If one already exists, just remove it.

```
# ln -s linux-2.4.21 linux
```

Now, if you already have a kernel .config for an older version, you would copy it into the new kernel directory and do this:

```
# make oldconfig
```

That will go through the config file and prompt you for a yes/no/help on any new options in the kernel. This is not very useful the first time around, but you will come to love this feature when new kernels come out and you dont want to go searching through to find new options.

Since you probably don't have a .config already, you will want to config from scratch. This is where it gets fun. cd into the kernel source directory(/usr/src/linux-2.4.21) and type:

```
# make menuconfig
```

Optionally, you can use "# make xconfig" and that will bring up a GUI configuration almost identical to the ncurses based one that I will be going over. I'm not covering the xconfig option because not everyone runs X on their box. So this way everyone can use it. Unless of course they dont have ncurses! But who doesn't right?

Okay, as daunting as this seems the first time, it will be fairly easy. It just takes time and you have to know what is in your system. The first option you see will be "Code maturity level options." This section has an option to prompt for development and/or incomplete code/drivers. You will want to enable this. It may warn you about it being unstable and such, but there will be additional warnings if the code will mess things up. It may not work, but it shouldn't hurt. You will also get lots of cool new options with this enabled.

The next section is all about modules and that stuff. You want to enable modules for sure. Some people like to have everything as modules. Some people like to have everything compiled into thier kernel. Personally, I like to compile everything as a module untill I know it is working properly. Then I will usually compile it into the kernel and ditch the module.

The next section is Processor type and features. One thing you want to set for sure is the first option. Just hit enter on the very top selection and it will bring up a window with a list of different supported processor types. Pick yours so the kernel will be optimized to run on your procesor. This is the first section with lots of confusing looking options. If your not sure if something should be enabled just read the help section on it. I recommend reading the help section on every single option in the kernel. It is very time consuming, but you will know exactly what you need afterwards. Another important thing in this section is SMP, or symmetric multi-processing support. If you need, it enable it. Otherwise, disable it. I have had quite a few problems enabling SMP with only one processor even though it's not supossed to cause problems.

The next section is fairly generalized options. Most things are straight forward and the defaults are usually okay. If you know you don't need something you can disable it to save space, like ISA bus support. It comes enabled by default, but most modern computers don't use the ISA bus, including mine, so I disable it. Again, if your not sure about an option read the help section on it. Another important option here, if you are running a laptop, is support for hot-pluggable devices and PCMCIA support. Also, you may or may not want to enable APM.

The next section is Memory Technology Devices. If you need this enabled you would know it. I have never had to use it. It's mainly for solid state storage on embeded devices, so you most definitely won't need it.

The next option is fairly straight forward. If you use the parallel port on the back of your computer, you need to enable this. Certain hardware requires some of the lower level options that come up when you enable it to be enabled as well. You would have to look that up though.

This next section is for Plug and Play, or more commonly known as Plug 'n Pray. If you use ISA, chances are you will want this too. If you aren't using the ISA bus, you wont need this option most likely. There is always that one obscure piece of hardware someone has from like 10 years ago that needs wierd options. You can usually find information on things like that on goodle.com/linux fairly easily.

Some of the important options in the next section, block devices, are floppy disk support. Most people still use floppys! Also, if you want to be able to mount .iso files, you can enable the loopback device option. Depending on your configuration, you might want RAM disk support. Again, I cant stress the importance of reading the help files to see what you need!!!

If you plan on using RAID or LVM (Logical Volume Management), you will want to check out this section. I have never used RAID before or LVM, so I dont know much about this section. I'm sure there are MANY howto's on the subject.

The next section, Networking Options, is fairly important, although the defaults are fine for most people. There are lots of different TCP/IP options that you may or may not want to enable. Also, if you have an Appletalk network, you might want to read through those options. I also believe you have to enable 802.11b or wireless networking seperately in this section, but its never come up for me.

This next section is another one of those obscure sections you probably won't need. If you have a telephony card (yeah, Ive never heard of one either), you can enable this and do VoIP stuff at the hardware level!

This next section seems small at first but is VERY important if you want your kernel to boot the first time around. You obviously want IDE support unless you are running an all SCSI system, which I have no experience with, so you might still need IDE! The second option, once expanded, has a ton of options. Some of the more important ones are DMA and other options for booting from add-on cards such as promise RAID cards. The option to boot off-board chipsets first would be important if for some reason your onboard chip is fried or you have to put the boot loader on a hard drive on a PCI IDE card. Also, you will want to read through ALL of the chipset support options. If you have one of these chipsets thats listed and you don't enable it, chances are your system will not boot properly.

This next section is for SCSI support. Most boxes wont need this but again some old obscure hardware thats NOT SCSI still uses the SCSI bus. The most common piece of hardware I can think of for this is the parllel Zip drives from Iomega. I happen to have one of those and they work great, but you will have to dig through this section for the ppa driver. As I said before, I have never used a SCSI device, but from what I understand most SCSI hardware has its own low level SCSI driver and there aren't many generic drivers. You will probably want to google for a howto about your hardware and that should help you a lot. I'm sorry I can't elaborate on this subject more because I'm sure a lot of people use SCSI devices.

The next two sections are for more obscure hardware! Damn, there's tons of support in the linux kernel, huh?! This appears to be for the LSI Logic Fusions Message Passing Technology devices, and the second is for I2O (Intelligent Input/Output) architecture. If you need these options you will most likely know so!

This next section is also fairly important if you want networking, and who doesn't? Most people will have a 10/100mb ethernet card. Almost all network cards are supported. Figuring out which driver you need is something else all together. But if you know what module your current kernel uses that will help a lot. You also might have a gigabit ethernet card and you can enable that here too. Also, I believe wi-fi cards are in this section as well as PCMCIA networking cards.

This next section is for ham radio support. I haven't used this buta lot of you might be interested in this.

Next section is for infared support. I have also never used this, but if you run a laptop you might want to work with this. I can't imagine it would be a top priority though.

This next section is new to me in 2.4.21. I dont remember ever seeing this section, but I have never even bothered reading about ISDN, so I might have just passed it over a lot. If you have ISDN, you might want to look into this section.

Next is more obscure hardware support. If you have a super old non-IDE, non-SCSI cdrom, this is where you would find the support for it.

The next section says it is for "USB Human Interface Device (HID) support." I would assume this means things like the retina scanners and fingerprint scanners, but I could be way off-base there. It's not really explained.

This next section is my favorite. This is where they put all the cool stuff. There are the default selections and then there is i2c, which you might want to enable depending on the version on lm_sensors you use. As of 2.4.0, i2c is a seperate package, at least in my distro. Also, if you have a non-PS2 mouse, you will have to enable it here. By the way, this doesn't include USB mice, which are a PAIN! Those are covered in the USB section. If you want to hookup a joystick for your hours of tuxracer playing, you would do it in here. There are lots of other chipset fixes and feature additions in here. Search through it for your chipset!!! Also, if you want to use an AGP card, you enable that in here too (as well as DRM). There are lots of other cool options for lots of cool hardware, so read through the help files in here for a while.

If you have a TV tuner card you can enable Video for Linux in this section. Watching TV on your linux box is kewl!

FILE SYSTEMS!!! File systems are so much fun. You will probably want things like ext2 and ext3 and possibly vfat. You might also use rieserfs or xfs or some other obscure file system! Also, you will want to enable the smb file system if you plan on mounting Windows shares from other Windows boxes. Also, NFS is another commonly used file system. If you need other file systems, you should know.

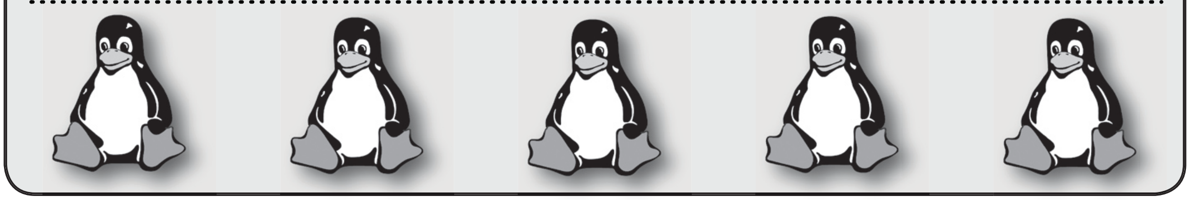
This is my second favorite section. I can't describe how pretty a frame buffered console looks. Not to mention if you get around to applying the console splash patch from SuSE and can set a background image on your console.

Everyone wants sound. This is where you would enable it. Personally, I use alsa drivers and not OSS, but some cards don't work well with alsa. Either way, you would want to enable sound support and possibly a driver for your card. If you don't know which driver your sound card uses, ***http://google.com/linux*** it!

** *SCREAM* USB!!!** I have never ever ever used a USB device. When I started out with Linux all I heard about was how much hell USB was. I understand that it is MUCH better now, but I can't really offer any advice on configuring the USB section aside from READ THE HELP FILES!@#. It might sound a little preachy by now, but I'm telling you, you will have a much smoother ride if you spend the time to read the help files.

The last section I know anything about is Bluetooth support. From what I have read, Bluetooth is VERY cool when you get it working. I really want to start messing around with this stuff, but haven't been able to get my hands on some Bluetooth hardware.

The last two sections you won't need help on if you need to use them. They are for writing your own kernel modules and stuff like that. Once you are done with all of that, just hot exit on the main menu and it will prompt you to save it. Obviously you won't want to do that all again so save it!!! I'm not going to go through how to compile the kernel once its configured because there are so many howtos about it and it is also different for most distributions. If you have any questions or comments or "you're a retard" emails, send them to ***snoogans@qwest.net***.



PERL CORNER

Hacking CoinStar

by ntheory

CoinStar is a network of “self-service coin counting machines located at the front entrances of leading supermarkets nationwide” according to their corporate website. CoinStar machines will count your change and spit out a receipt telling you how much money you tossed in (minus their service charge which is currently 8.9% in the US). Take the receipt to the register and they’ll give you cash so you don’t have to count and roll your change.

When CoinStar started several years ago a few friends and I thought it was a great way to annoy the cashiers we didn’t like. We’d go to the machine, throw a penny in, go up to the register with the receipt, and get the penny back from a very irritated cashier. It wasn’t until this year that I realized I could peer into how it worked and see what was behind it and how it might be vulnerable.

I went out to the supermarket and started the very slow, methodical process of reverse engineering the receipts. I threw in a penny and got the receipt, then I did it again, then with two pennies, three pennies... by the end of it I had about 10 - 15 receipts that were just loaded with data I could use.

When I got home I looked at the receipts and noticed several things. First I noticed (obviously) the barcode. I scanned the receipt for numbers that matched the numbers in the barcode and I found the following:

- The first three digits were always “040”
- The next four digits are the transaction ID (this is also located on the bottom of the receipt)
- The next five digits represent the amount of money you put in in pennies
- The last digit looked like it could be a checksum digit of some sort

I couldn’t figure out the pattern in the last digits so I hopped online and started looking up barcode formats. After a few minutes I came across a webpage that had a very in depth explanation of EAN-13 (http://www.barcodeisland.com/ean13.phtml). EAN-13 is the barcode format used in the US and Europe in most retail stores. The format is very well defined and almost every barcode scanner you’ll see today can read them. A normal EAN-13 barcode is broken up into four sections:

- Number system	(first two digits)
- Manufacturer code	(next five digits)
- Product code	(next five digits)
- Check digit	(last digit)

Now I had enough information to generate that last digit so I wrote a script to generate the check digit for a receipt and tried it on my CoinStar data. To my surprise the first receipt didn’t match and neither did the second. I realized that CoinStar borrowed the EAN-13 specification but broke the check digit scheme by implementing their own. ::sigh:: Something else to reverse engineer...

Let’s take a look at one of my one penny barcodes:

```
0 409705 000017
A BCDEFG HIJKLM
```

AB is the number system (04 in this case), CDEFG is the manufacturer code (09705, therefore the transaction number could possibly be five digits), HIJKL is the product code (00001 representing 1 penny), and M is the check digit. Now normally the EAN-13 check digit verification scheme goes like this:

1) Add all the even and odd digits before the check digit seperately
0 + 0 + 7 + 5 + 0 + 0 = The sum of all even digits (12)
4 + 9 + 0 + 0 + 0 + 1 = The sum of all odd digits (14)

2) Add the odd digit sum (multiplied by three) to the even digit sum
12 + (14 * 3) = 54

3) Divide it by 10 and take the remainder
54 mod 10 = 4

4) Subtract the remainer from 10
10 - 4 = 6

Unfortunately our check digit doesn't match. The uber-leet technicians at CoinStar made humongous changes to the algorithm to thwart cracking attempts. Here's what they did:

1 - 3) Same as above
4) Subtract the remainer from 11
11 - 4 = 7 == M (Success! This has been verified on many receipts of course)

This was a huge mistake on the part of CoinStar. The biggest problem is that EAN-13 barcode readers cannot read their format. They will spit back an error every time you try to scan it. I realized this when I redeemed one of my receipts and the cashier didn't scan it. Because of this the check is done by hand (and they won't bother calculating the check digit). The people verifying the receipt trust it completely. To make matters worse the CoinStar machines apparently aren't even hooked into the cash register system at all. They should've probably just set it up so you could enter the transaction number to make sure people are giving you fake receipts but they didn't.

Security through obscurity really backfired here and CoinStar is to blame. Maybe some scanners can read this special CoinStar format but I'm not sure who has them. What makes it all even worse is that CoinStar owns all of the machines and gives a small percentage of the percentage they take from you to the store that hosts the machine (confusing, eh?). The broken barcode scheme now places the liability solely on CoinStar since the stores can't easily check to make sure people's receipts are for real. Hopefully they'll wise up and fix this soon either by going to EAN-13 and/or hooking up to the register, or by distributing CoinStar barcode readers with the machines.

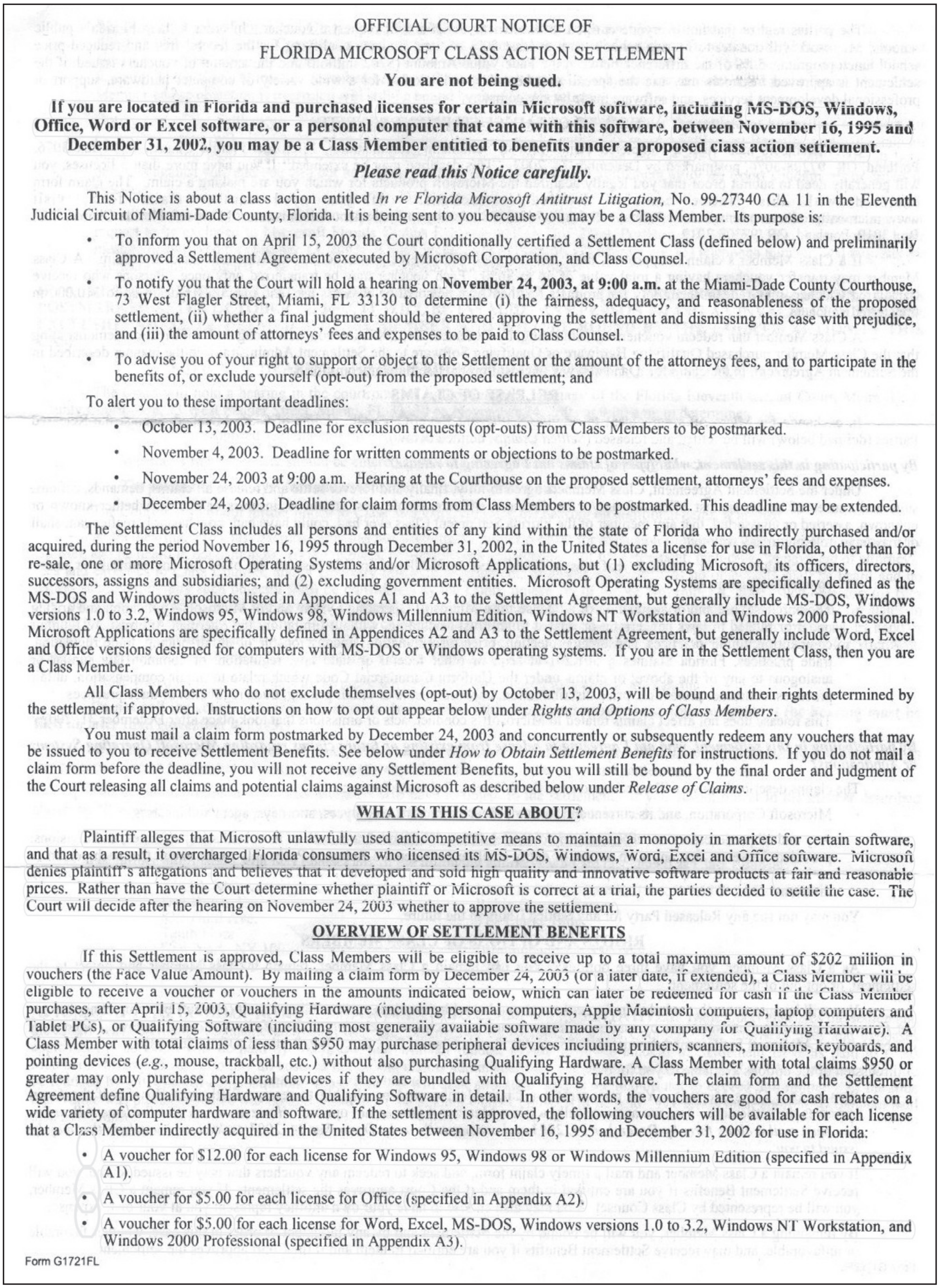
To finish my brief introduction to CoinStar's barcoding system I've attached a Perl script that will generate a CoinStar barcode from the first 12 digits (everything but the check digit). It makes use of ImageMagick to create PNGs of the barcode and it will tell you how much money the receipt is worth. By replacing the function that calculates the check digit with one that generates EAN-13 check digits it will also spit out valid EAN-13 barcodes. However I don't suggest you try to rip off a supermarket with this because I'm sure that they'll get suspicious when you come up with a dozen receipts for $999.99 (the maximum amount you can put in).

<DISCLAIMER =" Hacking Coinstar\nby ntheory" CODE="GenerateBarcode.pl"
DATE="6.30.03">
It takes much more than running this script to use this information for nefarious purposes.
You are responsible for your own actions. **This is not a tool for fraud.**
</DISCLAIMER>

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

# GenerateBarcode.pl
# by ntheory
#
#   Generates barcodes and the CRC digit for CoinStar receipts.
#
# June 26th, 2003 - Started development.
# June 30th, 2003 - After a few day haitus I resumed work and finished the
#                     code.

# Notes: Put the barcodes in a text file and redirect the file into STDIN.
#        You should only have the 12 digits of the barcode on each line and
#        nothing else.  Alternatively you can enter them by hand.

use Image::Magick;

# Set up some constants that we’ll need in the binary string generation phase.
$StartEndSentinel         = “101”;    # This goes at the beginning and end.
$CenterGuardPattern       = “01010”;  # This goes in the middle.
$StartEndSentinelLength   = length ($StartEndSentinel);
$CenterGuardPatternLength = length ($CenterGuardPattern);

# These are just defined for readability purposes.
$OddParity              = 0;
$EvenParity             = 1;
$ManufacturerCodeLength = 5;
$WholeLeftSideLength    = $ManufacturerCodeLength + 1;
$RequiredInputLength    = 12;
$RightSideLength        = 5;
$WholeRightSideLength   = $RightSideLength + 1;
$BarcodeHeight               = 50;
$SingleDigitLength      = 7;

# Get the parity map, the left hand coding table, and the right hand coding
# table.
$ParityMap       = GetParityMap ();
$LeftHandCoding  = GetLeftHandCodingTable ();
$RightHandCoding = GetRightHandCodingTable ();

$BarcodeCounter = 0;

# We’ll do this in a loop so you can generate many barcodes in a row.
while (<>) {
  # Get the unprocessed data and remove the trailing newline.
  $Unprocessed = $_;
  chomp ($Unprocessed);

  # Check to make sure that it makes sense.
  if (LooksValid ($Unprocessed)) {
    # Now convert it to a string of binary digits using some EAN-13 magic.
    # (http://www.barcodeisland.com/ean13.phtml)
    $Processed = $StartEndSentinel;

    # Get the first digit (determines the parity of the manufacturer code).
    $FirstDigit = int (substr ($Unprocessed, 0, 1));

    # Encode the second digit (always odd parity).
    $Offset       = 1;
    $CurrentDigit = int (substr ($Unprocessed, $Offset, 1));
    $Processed   .= $LeftHandCoding [$CurrentDigit][$OddParity];

    # Next encode the manufacturer code.
    for ($Loop = 0; $Loop < $ManufacturerCodeLength; $Loop++) {
      # Move to the next character.
      $Offset++;

      # Get the parity.
      $CurrentParity = substr ($ParityMap [$FirstDigit], $Loop, 1) eq “0” ? $OddParity
: $EvenParity;

      $CurrentDigit = int (substr ($Unprocessed, $Offset, 1));
      $Processed   .= $LeftHandCoding [$CurrentDigit][$CurrentParity];
    }

    # Slap the center guard pattern in there.
    $Processed .= $CenterGuardPattern;

    # Encode the right hand side.
    for ($Loop = 0; $Loop < $RightSideLength; $Loop++) {
      # Move to the next character.
      $Offset++;

      $CurrentDigit = int (substr ($Unprocessed, $Offset, 1));
      $Processed   .= $RightHandCoding [$CurrentDigit];
    }

    # Finally encode the check digit and slap the end sentinel in there.  We also tack
    the check digit
    # onto the unprocessed string so we can draw it onto the image below in a loop.
    $CheckDigit   = CalculateCheckDigit ($Unprocessed);
    $Unprocessed .= $CheckDigit;
    $Processed   .= $RightHandCoding [$CheckDigit];
    $Processed   .= $StartEndSentinel;

    # Now generate the image.
    $XOffset         = 100;
    $YOffset         = 100;
    $BarcodeLength   = length ($Processed);
    $BarcodeImage    = new Image::Magick;
    $BarcodeGeometry = ($BarcodeLength + $XOffset * 2) . “x” . ($BarcodeHeight + $YOff
    set * 2);

    $BarcodeImage->set  (size=>$BarcodeGeometry);
    $BarcodeImage->Read (“gradient:white-white”);
    # Don’t ask me why I always do this.

    # Draw the barcode.
    $YMin = $YOffset;
    $YMax = $YOffset + $BarcodeHeight;

    for ($Loop = 0; $Loop < $BarcodeLength; $Loop++) {
      $X            = $Loop + $XOffset;
      $PointsString = “$X,$YMin,$X,$YMax”;
      $StrokeColor  = substr ($Processed, $Loop, 1) eq “0” ? “White” : “Black”;
      $BarcodeImage->Draw (primitive=>”Line”, points=>”$PointsString”,
stroke=>”$StrokeColor”);
    }

    # Add in the digits below the barcode.
    # First clear out some space for the left digits.
    $BoxUpperLeftX  = $StartEndSentinelLength + $XOffset;
    $BoxUpperLeftY  = $BarcodeHeight - 10 + $YOffset;
    $BoxLowerRightX = $BoxUpperLeftX + $SingleDigitLength * $WholeLeftSideLength;
    $BoxLowerRightY = $BarcodeHeight + $YOffset;

    $PointsString = “$BoxUpperLeftX,$BoxUpperLeftY,$BoxLowerRightX,$BoxLowerRightY”;
    $BarcodeImage->Draw (primitive=>”Rectangle”, points=>”$PointsString”, fill=>”White”, 
stroke=>”White”);

    # Draw the left side digits
    $Y = $YOffset + $BarcodeHeight;

    for ($Loop = 1; $Loop < ($WholeLeftSideLength + 1); $Loop++) {
      $X = $BoxUpperLeftX + 1 + (($Loop - 1) * $SingleDigitLength);
      $Character = substr ($Unprocessed, $Loop, 1);
      $BarcodeImage->Annotate (text=>$Character, pointsize=>11, antialias=>”true”, 
x=>$X, y=>$Y, fill=>”Black”);
    }

    # Then clear out some space for the right digits.
    $BoxUpperLeftX  = $BoxLowerRightX + $CenterGuardPatternLength;
    $BoxLowerRightX = $BoxUpperLeftX  + ($WholeRightSideLength * $SingleDigitLength) 
- 1;

    $PointsString = “$BoxUpperLeftX,$BoxUpperLeftY,$BoxLowerRightX,$BoxLowerRightY”;
    $BarcodeImage->Draw (primitive=>”Rectangle”, points=>”$PointsString”, fill=>”White”, 
stroke=>”White”);

    # Draw the right side digits.
    for ($Loop = 1; $Loop < ($WholeRightSideLength + 1); $Loop++) {
      $X         = $BoxUpperLeftX + (($Loop - 1) * $SingleDigitLength);
      $Character = substr ($Unprocessed, $Loop + $WholeRightSideLength, 1);
      $BarcodeImage->Annotate (text=>$Character, pointsize=>11, antialias=>”true”, 
x=>$X, y=>$Y, fill=>”Black”);
    }

    # Draw the first very first digit.
    $X = $XOffset - 12;
    $Character = substr ($Unprocessed, 0, 1);
    $BarcodeImage->Annotate (text=>$Character, pointsize=>11, antialias=>”true”, x=>$X, 
y=>$Y, fill=>”Black”);

    $BarcodeCounterString = sprintf (“%03d”, $BarcodeCounter);
    $BarcodeImageName = “CoinStar-$BarcodeCounterString.png”;
    $BarcodeImage->Write ($BarcodeImageName);
    $BarcodeCounter++;

    # Let the user know that something happened.
    $TransactionID = substr ($Unprocessed, 3, 4);
    $Amount        = sprintf (“\$%3d.%02d”, int (substr ($Unprocessed,  7, 3)),
                                            int (substr ($Unprocessed, 10, 2)));
    print “Barcode generation for transaction $TransactionID ($Amount) was successful. 
The image was stored as $BarcodeImageName.  The CoinStar check digit was $CheckDigit.\
n”;
  }
}

sub LooksValid {
  $Data = $_ [0];
  $ReturnValue = 0;

  if (length ($Data) != $RequiredInputLength) {
    print “Wrong number of characters.
  $RequiredInputLength characters are needed to generate a barcode.\n”;
  }
  elsif ($Data =~ m/[^0-9]/) {
    print “There was a non-numeric character in your data.  Only numeric data is
accepted.\n”;
  }
  else {
    $ReturnValue = 1;
  }

  return $ReturnValue;
}

sub CalculateCheckDigit {
  $Data = $_ [0];

  # Coinstar really exhausted their technician’s with this one.  Below the code
  # may look very similar to a typical EAN-13 checksum calculation but it has
  # a surprise ending...

  $Sum = 0;

  # Do the weighted sum.
  for ($Loop = 0; $Loop < $RequiredInputLength; $Loop++) {
    $CurrentDigit = int (substr ($Data, $Loop, 1));

    # Even digits are added to the sum normally while odd digits are multiplied
    # by three before they’re added.
    if ($Loop % 2 == 0) {
      $Sum += $CurrentDigit;
    }
    else {
      $Sum += ($CurrentDigit * 3);
    }
  }

  # Mod the sum by 10.
  $Sum = $Sum % 10;

  # Normally here we’d subtract the sum from 10 but CoinStar had to be
  # different.  CoinStar has a Spinal Tap fetish (“We’ve got 11”).
  $CheckDigit = 11 - $Sum;

  # Make sure the check digit is less than 10.
  $CheckDigit = $CheckDigit % 10;

  return $CheckDigit;
}

sub GetParityMap {
  # This table tells us how to code the manufacturer’s code.
  my $ParityMap;

  $ParityMap [0] = “OOOOO”;
  $ParityMap [1] = “OEOEE”;
  $ParityMap [2] = “OEEOE”;
  $ParityMap [3] = “OEEEO”;
  $ParityMap [4] = “EOOEE”;
  $ParityMap [5] = “EEOOE”;
  $ParityMap [6] = “EEEOO”;
  $ParityMap [7] = “EOEOE”;
  $ParityMap [8] = “EOEEO”;
  $ParityMap [9] = “EEOEO”;

  return $ParityMap;
}

sub GetLeftHandCodingTable {
  # This table gives us the binary representation of the left hand digits.
  my $LeftHandCodingTable;

  $LeftHandCoding [0][$OddParity] = “0001101”;
  $LeftHandCoding [1][$OddParity] = “0011001”;
  $LeftHandCoding [2][$OddParity] = “0010011”;
  $LeftHandCoding [3][$OddParity] = “0111101”;
  $LeftHandCoding [4][$OddParity] = “0100011”;
  $LeftHandCoding [5][$OddParity] = “0110001”;
  $LeftHandCoding [6][$OddParity] = “0101111”;
  $LeftHandCoding [7][$OddParity] = “0111011”;
  $LeftHandCoding [8][$OddParity] = “0110111”;
  $LeftHandCoding [9][$OddParity] = “0001011”;

  $LeftHandCoding [0][$EvenParity] = “0100111”;
  $LeftHandCoding [1][$EvenParity] = “0110011”;
  $LeftHandCoding [2][$EvenParity] = “0011011”;
  $LeftHandCoding [3][$EvenParity] = “0100001”;
  $LeftHandCoding [4][$EvenParity] = “0011101”;
  $LeftHandCoding [5][$EvenParity] = “0111001”;
  $LeftHandCoding [6][$EvenParity] = “0000101”;
  $LeftHandCoding [7][$EvenParity] = “0010001”;
  $LeftHandCoding [8][$EvenParity] = “0001001”;
  $LeftHandCoding [9][$EvenParity] = “0010111”;

  return $LeftHandCoding;
}

sub GetRightHandCodingTable {
  # This table gives us the binary representation of the right hand digits.
  my $RightHandCoding;

  $RightHandCoding [0] = “1110010”;
  $RightHandCoding [1] = “1100110”;
  $RightHandCoding [2] = “1101100”;
  $RightHandCoding [3] = “1000010”;
  $RightHandCoding [4] = “1011100”;
  $RightHandCoding [5] = “1001110”;
  $RightHandCoding [6] = “1010000”;
  $RightHandCoding [7] = “1000100”;
  $RightHandCoding [8] = “1001000”;
  $RightHandCoding [9] = “1110100”;

  return $RightHandCoding;
}
```

/*comments*/

Our closing comments this issue is some preliminary information on the settlement between the state of Florida and Microsoft. The full text of this document can be found online at the BinRev 1.2 page at ***http://www.binrev.com*** under magazine.



OFFICIAL COURT NOTICE OF
FLORIDA MICROSOFT CLASS ACTION SETTLEMENT

You are not being sued.

If you are located in Florida and purchased licenses for certain Microsoft software, including MS-DOS, Windows, Office, Word or Excel software, or a personal computer that came with this software, between November 16, 1995 and December 31, 2002, you may be a Class Member entitled to benefits under a proposed class action settlement.

Please read this Notice carefully.

This Notice is about a class action entitled *In re Florida Microsoft Antitrust Litigation,* No. 99-27340 CA 11 in the Eleventh Judicial Circuit of Miami-Dade County, Florida. It is being sent to you because you may be a Class Member. Its purpose is:

- To inform you that on April 15, 2003 the Court conditionally certified a Settlement Class (defined below) and preliminarily approved a Settlement Agreement executed by Microsoft Corporation, and Class Counsel.
- To notify you that the Court will hold a hearing on **November 24, 2003, at 9:00 a.m.** at the Miami-Dade County Courthouse, 73 West Flagler Street, Miami, FL 33130 to determine (i) the fairness, adequacy, and reasonableness of the proposed settlement, (ii) whether a final judgment should be entered approving the settlement and dismissing this case with prejudice, and (iii) the amount of attorneys' fees and expenses to be paid to Class Counsel.
- To advise you of your right to support or object to the Settlement or the amount of the attorneys fees, and to participate in the benefits of, or exclude yourself (opt-out) from the proposed settlement; and

To alert you to these important deadlines:

- October 13, 2003. Deadline for exclusion requests (opt-outs) from Class Members to be postmarked.
- November 4, 2003. Deadline for written comments or objections to be postmarked.
- November 24, 2003 at 9:00 a.m. Hearing at the Courthouse on the proposed settlement, attorneys' fees and expenses.
- December 24, 2003. Deadline for claim forms from Class Members to be postmarked. This deadline may be extended.

The Settlement Class includes all persons and entities of any kind within the state of Florida who indirectly purchased and/or acquired, during the period November 16, 1995 through December 31, 2002, in the United States a license for use in Florida, other than for re-sale, one or more Microsoft Operating Systems and/or Microsoft Applications, but (1) excluding Microsoft, its officers, directors, successors, assigns and subsidiaries; and (2) excluding government entities. Microsoft Operating Systems are specifically defined as the MS-DOS and Windows products listed in Appendices A1 and A3 to the Settlement Agreement, but generally include MS-DOS, Windows versions 1.0 to 3.2, Windows 95, Windows 98, Windows Millennium Edition, Windows NT Workstation and Windows 2000 Professional. Microsoft Applications are specifically defined in Appendices A2 and A3 to the Settlement Agreement, but generally include Word, Excel and Office versions designed for computers with MS-DOS or Windows operating systems. If you are in the Settlement Class, then you are a Class Member.

All Class Members who do not exclude themselves (opt-out) by October 13, 2003, will be bound and their rights determined by the settlement, if approved. Instructions on how to opt out appear below under *Rights and Options of Class Members.*

You must mail a claim form postmarked by December 24, 2003 and concurrently or subsequently redeem any vouchers that may be issued to you to receive Settlement Benefits. See below under *How to Obtain Settlement Benefits* for instructions. If you do not mail a claim form before the deadline, you will not receive any Settlement Benefits, but you will still be bound by the final order and judgment of the Court releasing all claims and potential claims against Microsoft as described below under *Release of Claims.*

WHAT IS THIS CASE ABOUT?

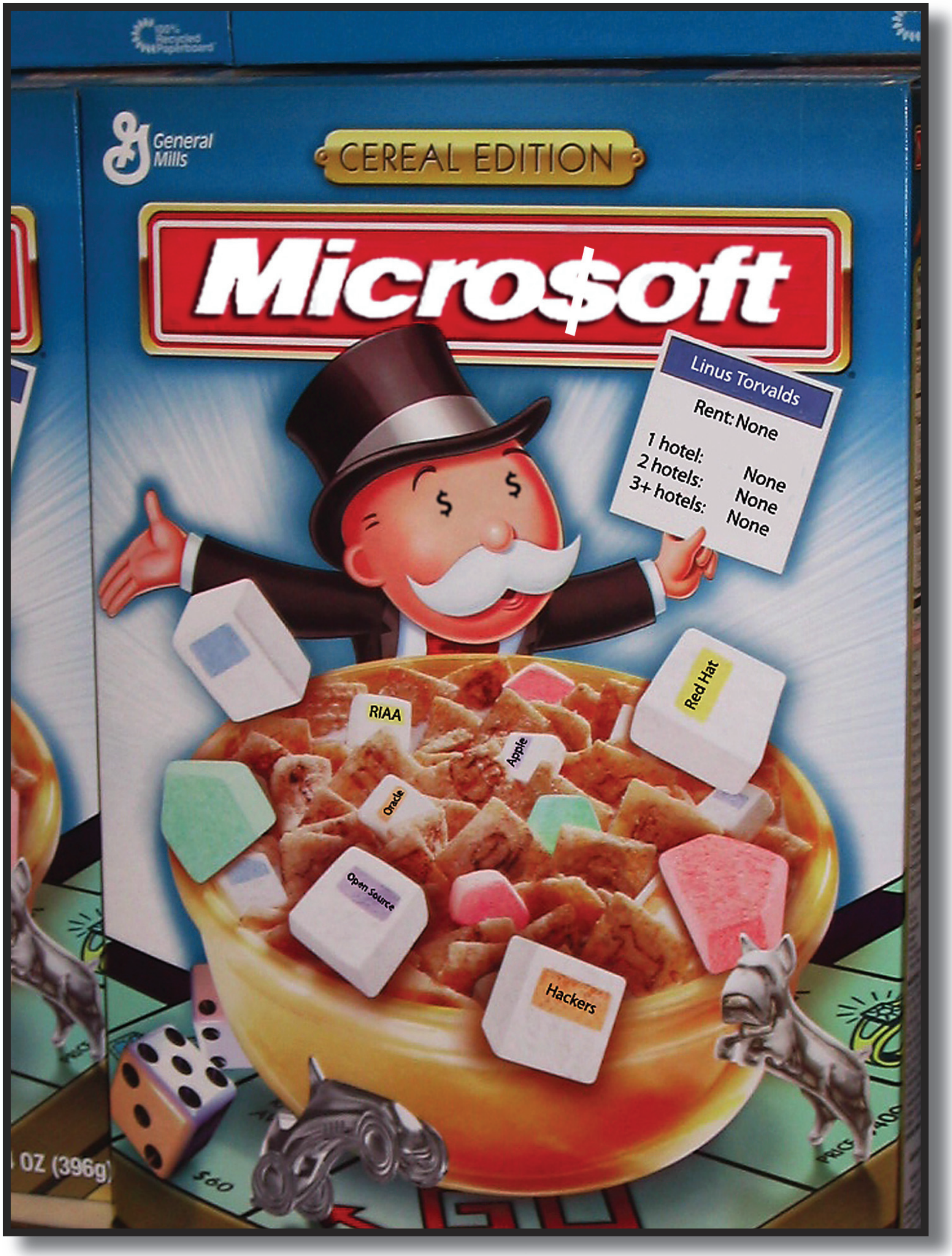
Plaintiff alleges that Microsoft unlawfully used anticompetitive means to maintain a monopoly in markets for certain software, and that as a result, it overcharged Florida consumers who licensed its MS-DOS, Windows, Word, Excel and Office software. Microsoft denies plaintiff's allegations and believes that it developed and sold high quality and innovative software products at fair and reasonable prices. Rather than have the Court determine whether plaintiff or Microsoft is correct at a trial, the parties decided to settle the case. The Court will decide after the hearing on November 24, 2003 whether to approve the settlement.

OVERVIEW OF SETTLEMENT BENEFITS

If this Settlement is approved, Class Members will be eligible to receive up to a total maximum amount of $202 million in vouchers (the Face Value Amount). By mailing a claim form by December 24, 2003 (or a later date, if extended), a Class Member will be eligible to receive a voucher or vouchers in the amounts indicated below, which can later be redeemed for cash if the Class Member purchases, after April 15, 2003, Qualifying Hardware (including personal computers, Apple Macintosh computers, laptop computers and Tablet PCs), or Qualifying Software (including most generally available software made by any company for Qualifying Hardware). A Class Member with total claims of less than $950 may purchase peripheral devices including printers, scanners, monitors, keyboards, and pointing devices (*e.g.*, mouse, trackball, etc.) without also purchasing Qualifying Hardware. A Class Member with total claims $950 or greater may only purchase peripheral devices if they are bundled with Qualifying Hardware. The claim form and the Settlement Agreement define Qualifying Hardware and Qualifying Software in detail. In other words, the vouchers are good for cash rebates on a wide variety of computer hardware and software. If the settlement is approved, the following vouchers will be available for each license that a Class Member indirectly acquired in the United States between November 16, 1995 and December 31, 2002 for use in Florida:

- A voucher for $12.00 for each license for Windows 95, Windows 98 or Windows Millennium Edition (specified in Appendix A1).
- A voucher for $5.00 for each license for Office (specified in Appendix A2).
- A voucher for $5.00 for each license for Word, Excel, MS-DOS, Windows versions 1.0 to 3.2, Windows NT Workstation, and Windows 2000 Professional (specified in Appendix A3).

Form G1721FL



submitted by: w1nt3rmut3

NEW! available exclusively on the LINUX WORLD TOUR

RECORD NO.
2.4.X
Southern Music
(ASCAP)
OPEN SOURCE RECORDS
STEREO
the
LINUX
TOUCH
music to compile by:
YOU, ME & KDE
COMPILE MY HEART
ROOT SWEET ROOT
WHAT VIRUS?
...AND MANY MORE!

Guaranteed 100% M$ Free High-Fidelity!