GIAC
CERTIFICATIONS

Global Information Assurance Certification Paper

Copyright SANS Institute
Author Retains Full Rights

This paper is taken from the GIAC directory of certified professionals. Reposting is not permited without express written permission.

Interested in learning more?

Check out the list of upcoming events offering
"Security Essentials (Security 401)"
at http://www.giac.org/registration/gsec

Practical demonstration of 802.11 wireless network system risk for non-technical business managers

Name: Mark Fromm
Certification: GIAC-GSEC
Version: 1.4b, option 1

Table of Contents:

I. Abstract

II. Introduction

III. System Summary

IV. Construction Details

V. System Operation

VI. Conclusion

VII. Futures

VIII. References

© SANS Institute 2004, Author retains full rights.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

I. Abstract

The deployment of wireless 802.11 networking often creates serious security and potential liability risk. Security experts understand wireless network risk, and understand how 802.11 networks can be compromised.

Non-technical corporate management often proposes wireless deployments for convenience, but do not understand the security implications of 802.11 wireless network deployments.

When the security expert attempts to explain the risk of 802.11 wireless networks to others, the risk issues often remain abstract due to the technical nature of the attack and compromise methods.

This paper will provide several demonstrations of 802.11 wireless system risk to non-technical staff, using special purpose, low cost honeypot systems to illustrate potential security vulnerabilities in a in a clear, easily understood manner.

II. Introduction

The risk factors associated with deploying and using 802.11 wireless networking systems are well understood for those who study computer security. Security experts are typically well versed in the weaknesses of wireless security measures such as WEP and MAC address restrictions and fully understands the various ways a wireless system may be attacked.

A security expert also follows “high profile” compromises of wireless systems reported in the computer security literature. Unfortunately, management and non-technical system users of most companies do not read computer security literature on a regular basis and seldom understands the various 802.11 system compromise methods. Descriptions of wireless network systems attacks and vulnerabilities result in a significant amount of jargon and acronyms, which make it difficult to understand the seriousness and magnitude of wireless system risk.

The number of 802.11 “hotspots” grows by the week. 802.11 wireless network interface cards and access points are sold in great numbers at consumer electronics retail outlets. The large number of 802.11 wireless system deployments by home users, businesses, and commercial hotspots masks the inherent risks these systems create. If everybody is doing it, it must be secure, right? 802.11 wireless networking equipment is sold with even the most basic security measures disabled for easy installation by anyone, regardless of technical background. The inability of most users of residential and commercial 802.11 deployments to even recognize that they have been compromised only adds to the problem. Weak or non-existent logging practices contribute significantly to the risk.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

In the article "Senior management slow to understand wireless risks'" customer engineering manager Steve Woolf maintains that

> "Senior managers typically have a limited understanding of the relationship between IT and business. They are the users, not the architects, of these high-tech systems and as such they aren't able to reconcile risk with return on investment" [1]

Examples of wireless system compromise with serious implications include the recent hijacking of a residential wireless internet connection to download "kiddie porn" in Canada as reported in the Security Focus article "Wi-Fi hacker caught downloading child porn".[2] There can be serious legal and reputation implications to a company that becomes the target of a federal child porn investigation due to unauthorized use of an open wireless 802.11 access point.

Another recent Wireless hacking incident occurred in Michigan, where two hackers that exploited a 802.11 wireless networking system accessed the internal wide area network of Lowe's Hardware stores nationwide, planting hacking software in various computer systems and capturing credit card information from point of sale terminals.[3] Currently, there are significant fines, legal liability, and public exposure via various state mandatory notification laws for a company when it fails to maintain customer credit card data by a secure method.

If a corporation's 802.11 wireless networking system is compromised during a weekend and a significant amount of illicit music trading happens over the company Internet link, the business could find itself in the crosshairs of the RIAA file sharing lawyers. The article "Corporate Liability for Online File sharing" notes that corporate networks are attractive targets for filesharing litigation due to their ease by which the companies responsible can be traced back through their IP address.[4]

> "With this information in hand, copyright holders could seek to bring actions against the corporate entity without ever knowing, or needing to know, the specific identity of the individual user on the other side of the corporate firewall."

---

[1] Wearden, Graeme. "Senior management 'slow to understand wireless risks'" ZDNet UK. November 20, 2003 http://news.zdnet.co.uk/communications/wireless/0,39020348,39118028,00.htm

[2] Leyden, John. "Wi-Fi hacker caught downloading child porn" The Register Nov 24 2003. http://www.securityfocus.com/news/7514

[3] Poulsen, Kevin. "Wireless hacking bust in Michigan". SecurityFocus Nov 12 2003. http://www.securityfocus.com/news/7438

[4] Luria, Mary M. and Kibel, Gary A. "Corporate Liability For Online File Sharing" Davis and Gilbert LLP http://articles.corporate.findlaw.com/articles/file/01009/009389

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

The 802.11 security issues are not just confined to those who set up an access point, but wireless users as well. Users of public 802.11 "hotspots" can face security issues as well. Using a new tool, Airsnarf,[5] hackers can disconnect legitimate users from a public hotspot. The attacker then displays a login page that appears to be the service provider's authentication screen. When users sign back on, their userIDs and passwords go to the attackers, not the wireless provider. [6]

Providing 802.11 networking can create legal liabilities for companies that deploy these networks in insecure configurations. In the article "Free Wi-Fi Hotspots and Potential Legal Problems" the author analyzes the various legal liability issues that will most certainly be present in future court cases. The author discusses the problem of "drive by warspamming" using unsecured 802.11 wireless networks, and says:

> "It is only a matter of time before a malicious hacker launches a major attack on a corporate or government computer while logged on anonymously in a city park, or some generous soul with a public home access point gets a visit from the FBI because someone e-mailed child pornography to every member of Congress from a car parked outside the samaritan's house" writes Jonathan I. Ezor. [7]

Unfortunately, non-technical business managers never sees nor reads articles such as the ones cited above, and therefore underestimate the risks and potential liabilities of wireless networking. Another reason for the understating of risk is that many wireless security incidents never get reported publicly. Due to reasons including damage to the companie's reputation, financial effects including stock price drop, and insurance rate increases, most companies are loath to admit they have been victim of a security breach.

For this reason, senior management, who intelligently and appropriately deal with risk management from a business perspective on a regular basis, apply flawed reasoning to assessing wireless security risk, and often approve or champion "security challenged" wireless deployments for the convenience 802.11 networking provides.

© SANS Institute 2004, Author retains full rights.

---

[5] Airsnarf - A rogue AP setup utility 0.2. The Shmoo Group http://airsnarf.shmoo.com/

[6] Brandt, Andrew. "A latte, a Wi-Fi link and a hacker" PC World Nov 5, 2003 http://www.computerworld.com/mobiletopics/mobile/story/0,10801,87523,00.html?f=x68

[7] Ezor, Jonathan I. "Free Wi-Fi Hotspots and Potential Legal Problems" BizLawTech - The Official Web site for the Institute for Business, Law and Technology at Touro Law Center Jun 2, 2003 http://iblt.tourolaw.edu/blog/archives/000010.html

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

Senior management seldom knows about the widespread popularity of "War Driving" by individuals, some with hostile intent. Non-technical managers do not recognize how easy it is to successfully compromise many corporate wireless systems. Due to this incomplete understanding of the information asset risk, senior management needs clear, non-technical information to demonstrate actual risk wireless systems create.

I became frustrated in my own attempts to convey 802.11 wireless security risk to non technical management. My descriptions of common, effective, and readily available tools to discover wireless access points such as netstumbler,[8] "kismet[9] and "bsd-airtools,[10] coupled with network enumeration tools such as nmap,[11] and techniques such as null sessioning were met with blank stares. My concerns over the ease of wireless network compromise were not adequately conveyed.

Conventional honeypot methodologies can be used to demonstrate 802.11 wireless network risk. However, conventional wireless honeypot technology is overkill for simply demonstrating risk, both in cost and complexity. This paper will detail the construction of a low cost portable wireless honeypot, which can run "standalone" from a battery, without AC power or networking for easy setup and flexible placement. This honeypot will appear to the attacker as a medium sized company network, containing what appears to be attractive, proprietary information on a Windows server. By capturing 802.11 link level events, I can demonstrate the frequency of "drive by NetStumblers". Furthermore, by demonstrating unauthorized remote Windows file services access, I can illustrate more clearly 802.11 wireless risk issues to a non-technical person. This honeypot will log all access and attacks for later analysis, and will serve to demonstrate risk factors of 802.11 wireless systems deployment in a way that can be clearly communicated to management and is free from jargon or technical terms. Honeypots are usually divided into two types; "high interaction" and "low interaction". Lance Spitzner provides a good distinction of these two types:

> "Low-interaction honeypots have limited interaction, they normally work by emulating services and operating systems"
> "High-interaction honeypots are different, they are usually complex solutions as they involve real operating systems and applications. Nothing is emulated, we give attackers the real thing" [12]

---

[8] NetStumbler.com – the world of WiFi http://www.netstumbler.com/

[9] Kismet is an 802.11 layer2 wireless network detector/sniffer http://www.kismetwireless.net/

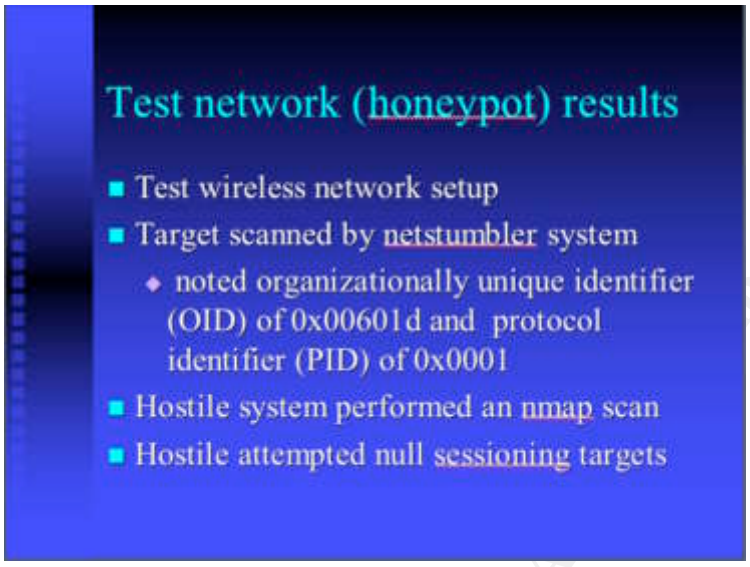
[10] bsd-airtools is a package that provides a complete toolset for wireless 802.11b auditing dachb0den labs http://www.dachb0den.com/projects/bsd-airtools.html

[11] Nmap ("Network Mapper") is a utility for network exploration, auditing http://www.insecure.org/nmap/

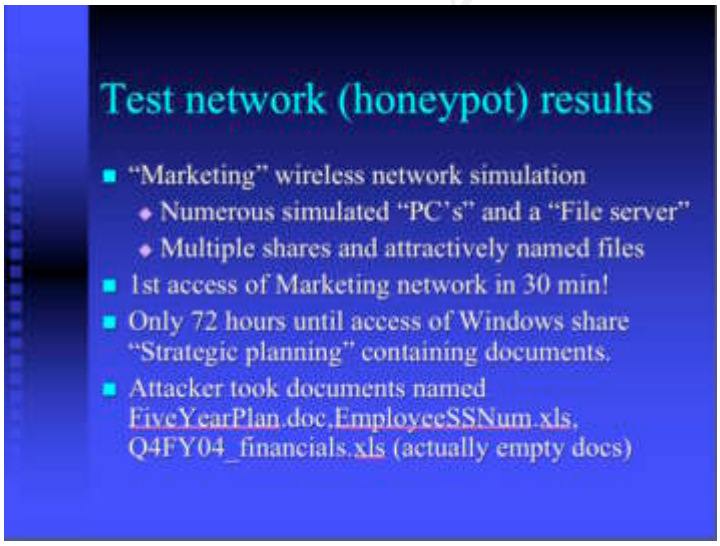
[12] "Honeypots - Definitions and Value of Honeypots" Lance Spitzner http://www.tracking-hackers.com/papers/honeypots.html

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

I wanted to construct a system that clearly demonstrated the ease and frequency of compromise to a typical 802.11 network system. Wireless honeypot solutions seemed ideal, but low interaction systems often do not provide information about the attacks that non-technical persons can understand. High interaction types are more complex and resource intensive. For this project, I was looking for a middle ground between the two designs. When presenting to management the risks of 802.11, would the following slide be easily grasped and effective?



Or, would this next slide be more successful in demonstrating risk of deploying 802.11 wireless networking systems?



In my opinion the second slide is definitely more effective in illustrating the potential risk of wireless network compromise to a non-technical executive.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

III. System Summary

The desire to show “file server access” which clearly illustrates true hostile intent to non-technical persons led me to build my honeypot system to incorporate actual SMB file services. I include a DHCP server to give “drive by clients” the convenience of a pre-configured IP address in the honeypot network, and DNS services which will appear to answer DNS queries. In addition, I can capture 802.11 link level headers and demonstrate the frequency of “drive by NetStumblers” and potential access point mapping.

I built the entire honeypot network and system on a laptop, creating a self contained wireless system and simulated network without any connections to production networks or the Internet to insure that compromise of the honeypot would not result in additional risk.

For hardware, I used older IBM 380ED Thinkpads, with 167Mhz Pentium CPU and 80 MB of RAM, coupled with wireless Prism chipset SMC 802.11b PCMCIA cards. This limited computer is not capable of effectively supporting more conventional methods of high interaction honeypot systems using VMware. Laptops of this hardware performance are readily available and inexpensive, often discarded as corporate systems due to their inability to support current Microsoft operating systems and applications. Deploying honeypots on older laptops is an efficient reuse of systems that would be otherwise discarded. Used 802.11 Prism 2 PCMCIA cards are readily and inexpensively available. There are also newer cards, based on Prism 2.5 and Prism 3 chipsets listed in the appendix that are also effective.

For operating system and software, I chose OpenBSD 3.4 for it’s “secure by default” configuration, with various add-on components and scripts to create the wireless honeypot system.

I typically deployed the machines for testing in a pair, with the first machine running the as an 802.11 Access Point by operating the card in hostAP mode. This first machine contained a DHCP server, SMB services for 20-50 “corporate LAN client machines” and a “file server”, and DNS services. The second machine was operated with its wireless card running in monitor only mode to log all 802.11 traffic. The advantages of operating a self contained laptop with an 802.11 PCMCIA card in hostAP mode, rather than using a conventional wireless 802.11 Access point include the following:

- Allowed me to see the “low level” 802.11 management frames and probe requests, which are not visible to users of low end commercial access points. Monitoring at this level can show reconnaissance efforts.
- Allowed my machine to serve the DHCP address and trigger actions based on new “drive by” client leases

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

- Provided “apparent” DNS services to clients which allows more information to be gathered regarding intended external destinations
- Easy to set up and relocate, can run for several hours without AC power
- Can be deployed longer term in a car using available cigarette lighter to laptop power converter. This allows easy monitoring for hostile activity from a company parking lot, which is where an attacker might start initial reconnaissance or attempted actual compromise of a company.

IV. Construction Details

Base System configuration

I performed a clean format on the IBM Thinkpad, and performed a fresh install of OpenBSD, which has had only one remote hole in the default install discovered in 7 years. Instructions for obtaining or installing OpenBSD can be found at the OpenBSD website.[13]

I did not install the X11 GUI packages. This resulted in a smaller install, and left me additional room for logging. I patched the source tree current with all security patches. I rebuilt the patched source segments and built a new kernel. It is important you apply current patches, which are supplied as source. [14]

I optimized the kernel by removing unneeded devices and increasing the number of NMBCLUSTERS. This kernel value affects size of the kernel mbuf cluster map. Due to the large number of virtual interfaces I was creating on my machine, I could exhaust these resources under heavy portscanning with nmap during my initial system testing on an isolated Ethernet network, which led me to increase this value. Note that these specific IBM laptops contain limited physical memory, which constrains the amount I can allocate to this memory pool.

A handy utility for displaying OpenBSD device maps is “dmassage”.[15] This tool lists detected devices in an easy to read tree format. This utility can be also be used to quickly and accurately comment out hardware devices in the OpenBSD GENERIC kernel configuration file that do not exist on the target system. In the past I have often made typos when commenting out unneeded items from GENERIC when stripping down OpenBSD kernel configurations or inadvertently removed critical interfaces. This can result in a kernel that will not boot.

---

[13] The OpenBSD FAQ, which covers obtaining and installing OpenBSD is located at http://www.openbsd.org/faq/index.html

[14] The up to date list of current OpenBSD patches, as well as the patches themselves, which are supplied as source code, is available at http://www.openbsd.org/errata.html

[15] The dmassage utility is available at http://www.sentia.org/projects/dmassage/

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

Fixing this is time consuming, as corrections must be made to the kernel config file, reconfiguration of the kernel build tree, and recompiling of the kernel. This cycle is more painful and time consuming on 167mhz Pentium target machines. I only used the dmassage scripts to display device trees and generate kernel configuration files, I did not use it to work directly on a kernel, which it can modify by using generated config –e –o (ukc>) commands.

On my systems, I disabled sendmail, ntpd and inetd by setting them to NO in /etc/rc.conf. For logging, based on the specific test, I used both the OpenBSD packet filter as well as tcpdump.

To use OpenBSD's packet filter, pf, you need to configure /etc/rc.conf to enable pf and pflogd. For those unfamiliar with pf, I suggest starting with two rules in /etc/pf.conf:

```
pass in  log all
pass out log all
```

This will at least get you started with logging until you become more familiar with pf capabilities and pf.conf syntax. You should consult the excellent manpages on pf.conf and pflogd. You also will probably want to set pflogd to use a larger snaplen than the default, so you can capture the entire packet, assuming you have enough disk space. The default pf logfile is /var/log/pflog, which is a tcpdump compatible format. You can also monitor the pf logging in real time by bringing up the pflog interface and attaching tcpdump to it:

```
ifconfig pflog0 up
tcpdump -n -e -ttt -i pflog0
```

Once I had the base operating system configured, I compiled and installed Samba from the ports tree, as my desire was to simulate "real" windows file servers containing proprietary data. I used the named and dhcpd that are included in the OpenBSD distribution.

I created a large number of 'alias" interfaces on the system using "ifconfig wi0 alias *addr* netmask *mask*" commands. Each alias represented an apparent windows client IP addresses and interfaces on my simulated LAN.

I wrote scripts to create Samba configuration files, as well as separate lock and log directories representing each client. This script also created the startup line with the specific IP alias the client was to bind to. A requirement for running multiple instances of Samba on a machine, bound to separate IP addresses is that each instance must have a unique NetBIOS name and use separate lock, log and PID directories. Startup of the multiple smbd and nmbd processes on the Pentium 167mhz was resource intensive but took surprisingly small amounts of system resources once running. See the system operations section for actual load and process information.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

My scripts also generated the bind configuration files and dhcpd.conf for the system. There are additional details on these scripts in the appendix. Due to the tedious nature of generating these configuration files correctly, my scripts greatly simplified changing IP address ranges, domain names and client machine names. Bind name services were deployed rather unconventionally, in that the service is configured to answer any resolution request given by a connected wireless client with a bogus reply. This was done to allow the attacker apparent name resolution capability, in order that additional data might be gathered on the nature of the attackers actions while connected to my system.

IMPORTANT: Since this system runs a self contained DHCP server, as well as a bogus name server, it is critical that a system like this is NEVER attached to a production Ethernet network. This system works great as an 802.11 wireless AP server. It would be nothing short of disastrous if connected to an Ethernet interface on a corporate LAN. Connecting a DHCP and bogus name server to a production network would be an incredibly stupid and very disruptive action, and you will incur the wrath of your network administrator. You have been warned!

Installation of the wireless AP functions

In OpenBSD, an 802.11 wireless card with a Prism chipset can be run in "hostAP" mode, which allows all 802.11 Access Point functions to be serviced via the OpenBSD system. This allows the creation of a complete, self-contained and standalone 802.11 network system.

There are specific requirements on hostAP mode supported cards and firmware versions. I initially obtained two SMC 2632W cards. Unfortunately, these SMC cards came with station firmware 0.7.6, which will not operate the card in hostAP mode. The SMC cards I used had platform type NICID 0x8002. Also note they are Prism 2 based and are not compatible with Prism 2.5 or Prism3 firmware, which the newer cards are designed to use.

The specific card firmware version can be ascertained by examining the dmesg output after boot with the card inserted.

```
wi0 at pcmcia1 function 0 "SMC, SMC2632W, Version 01.02" port 0x400/64: irq 3
wi0: PRISM2 HWB3163 rev.A, Firmware 0.3.0 (primary), 0.7.6 (station), address
00:e0:29:91:97:f1
```

I used a Windows based Intersil Prism upgrade utility on a separate Windows laptop along with the hex firmware file S10008c2.hex, which successfully upgraded the SMC cards to firmware version 0.8.2.[16]

[16] I found the Intersil Prism 802.11 card update utility at ftp://ftp.dlink.com/Wireless/DWL650/Firmware/

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

After the firmware update, dmesg on the OpenBSD laptop reported the card as:

```
wi0 at pcmcia1 function 0 "SMC, SMC2632W, Version 01.02" port 0x400/64: irq 3
wi0: PRISM2 HWB3163 rev.A, Firmware 0.3.0 (primary), 0.8.2 (station), address
00:e0:29:91:97:f1
```

Once the 802.11 card was running at firmware version 0.8.2 on the OpenBSD laptop, I was able to ifconfig the card into hostAP mode. I was also able to detect my new 802.11 hostAP mode traffic from a second OpenBSD laptop when it's card was placed in monitor mode. The system worked as an 802.11 access point, but I noticed very poor power output from the SMC card. I decided to try updating to a more current version of firmware for the SMC card.

I obtained firmware version 1.4.9 for Prism 2 cards compatible with my card type of NICID 0x8002. After the successful upgrade, I noticed approximately 30db greater transmit output when used in hostAP mode. After the final upgrade, I got the following 802.11 card related dmesg output:

```
wi0 at pcmcia1 function 0 "SMC, SMC2632W, Version 01.02" port 0x400/64: irq 3
wi0: PRISM2 HWB3163 rev.A, Firmware 0.3.0 (primary), 1.4.9 (station), address
00:e0:29:91:97:f1
```

For version 1.4.9 firmware for Prism 2 cards with NICID 0x8002, I used the hex file I found at linux.jinsun.net.[17]

The md5sum of the firmware file I used to upgrade my SMC cards is:

```
1e4ff039dc7309473dadde7c7ff12de9  s1010409.hex
```

If you are going to be flashing firmware on Prism cards to support hostAP mode, you should reference "Jun Sun's Mini-Howto on flashing Intersil Prism cards".[18]

You can find firmware updates for Prism 2.5 and Prism 3 chipset cards, which have HFA3842, NICID 0x800A and higher at Jun Sun's site and at netgate.com.[19]

[17] I found Prism 2 card 1.4.9 firmware at http://linux.junsun.net/intersil-prism/firmware/1.4.9/

[18] Jun Sun's Mini-howto on Flashing Intersil Prism Chipsets is at http://linux.junsun.net/intersil-prism/

[19] The main netgate site is at http://www.netgate.com/
The prism firmware is at http://www.netgate.com/support/prism_firmware/

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

V. System Operation

I wrote scripts to generate the configuration files to run multiple smbd processes on separate virtual interfaces, as well as generate the named configuration files and the dhcpd.conf. To create the configuration for one “server” and 25 apparent clients requires the creation of 79 directories and approximately 30 configuration files. There are a number of separate spool, lock and log files required to run 26 smbd/nmbd instances on one machine. Creating all these configuration files by hand is tedious and error-prone, which is why I developed the build scripts. My build scripts also creates a driver script to bring up the Ethernet interfaces on the wi0 802.11 ethernet card, starts all the smbd/nmbd processes, brings up the DHCP server and starts the named server for DNS resolution. There is additional information on my scripts in the appendix, and all scripts and test results will be posted to the website listed in the appendix.

The simulated “file server” has a number of network shares. Within these creatively names shares I have placed files with interesting names and bogus content. For example, I have files named EmployeeSSNumbers.xls and Q1_FY04_financials.xls. I set owner on these to root, and share it out read only, allowing an 802.11 remote attacker to retrieve these files.

On the “simulated clients”, I only have a C$, or hidden administrative share, which I export writable by guest. Within the client samba configuration, I set the location of the C$ share for all clients to a single location, /var/samba/C_dollar.

To provide “content” for the C$ share, I burned a copy of a freshly installed Windows 98 “C:” drive to a CD-R. I mount the CD-R, then union mount it to /var/samba/C_dollar.

```
chmod a+w /var/samba/C_dollar
mount_cd9660 /dev/cd0c /mnt
mount -t union -o -b /mnt /var/samba/C_dollar
```

This makes it appear that the C: drive of the clients is remotely manageable. I am curious if an attacker decides to perform additions or creative changes to files located within the client share points. I can easily drop the union mount and discover what debris has been left on the apparent client C: drive filesystem.

To bring up the 802.11 card initially in hostAP mode, I used the following:

```
#!/bin/sh
ifconfig wi0 inet up nwid Marketing media DS11 mediaopt hostap
wicontrol -f 6
wicontrol -m c:0:f:f:e:e
ifconfig wi0 192.168.55.1 255.255.255.0
```

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

This sets the SSID to “Marketing”, the channel to 6, the speed to 11mb, enables hostAP mode, sets a new MAC address and configures an IP address on the interface. For channels, I suggest one of the three non-overlapping frequencies of 1, 6 or 11 for 802.11 cards sold and used in the US.

The result if running the previous commands is:

```
wi0: flags=8943<UP,BROADCAST,RUNNING,PROMISC,SIMPLEX,MULTICAST> mtu 1500
        address: 0c:00:0f:0f:0e:0e
        nwid Marketing
        powersave off
        media: IEEE802.11 DS11 hostap
        status: active
        inet6 fe80::2e0:29ff:fe91:97f1%wi0 prefixlen 64 scopeid 0x1
        inet 192.168.55.1 netmask 0xffffff00 broadcast 255.255.255.0
```

I brought up the 802.11 interface prior to running the scripts that started the smbd/nmbd, DHCP and named processes. Note that in my testing, I changed the MAC address on startup of the interface from **`00:e0:29:91:97:f1`** to something I could easily remember and spot in network captures: **`c:0:f:f:e:e`**

I set the cards MAC address to **`c:0:f:f:e:e`** because it is an easy hex string for me to remember, and also to conclusively answer the statement I keep hearing that “You can secure an 802.11 wireless link by restricting it to known MAC addresses” As this clearly illustrates, I can set the MAC address of my wireless cards to anything I want it to be. Furthermore, when using BSD based wireless tools such as dstumbler,[20] the system can be configured to change the cards MAC address to random values every few seconds.

Once the wireless interface is up and running, and the virtual interfaces for the “apparent client” machines are running, an ifconfig produces the output shown on the following page:

[20] dstumbler is part of the bsd-airtools utilities http://www.dachb0den.com/projects/dstumbler.html

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

```
Ifconfig wi0
wi0: flags=8943<UP,BROADCAST,RUNNING,PROMISC,SIMPLEX,MULTICAST> mtu 1500
        address: 0c:00:0f:0f:0e:0e
        nwid Marketing
        powersave off
        media: IEEE802.11 DS11 hostAP
        status: active
        inet6 fe80::2e0:29ff:fe91:97f1%wi0 prefixlen 64 scopeid 0x1
        inet 192.168.55.1 netmask 0xffffff00 broadcast 255.255.255.0
        inet 192.168.55.201 netmask 0xffffff00 broadcast 192.168.55.255
        inet 192.168.55.202 netmask 0xffffff00 broadcast 192.168.55.255
        inet 192.168.55.203 netmask 0xffffff00 broadcast 192.168.55.255
        inet 192.168.55.204 netmask 0xffffff00 broadcast 192.168.55.255
        inet 192.168.55.205 netmask 0xffffff00 broadcast 192.168.55.255
        inet 192.168.55.206 netmask 0xffffff00 broadcast 192.168.55.255
        inet 192.168.55.207 netmask 0xffffff00 broadcast 192.168.55.255
        inet 192.168.55.208 netmask 0xffffff00 broadcast 192.168.55.255
        inet 192.168.55.209 netmask 0xffffff00 broadcast 192.168.55.255
        inet 192.168.55.210 netmask 0xffffff00 broadcast 192.168.55.255
        inet 192.168.55.211 netmask 0xffffff00 broadcast 192.168.55.255
        inet 192.168.55.212 netmask 0xffffff00 broadcast 192.168.55.255
        inet 192.168.55.213 netmask 0xffffff00 broadcast 192.168.55.255
        inet 192.168.55.214 netmask 0xffffff00 broadcast 192.168.55.255
        inet 192.168.55.215 netmask 0xffffff00 broadcast 192.168.55.255
        inet 192.168.55.216 netmask 0xffffff00 broadcast 192.168.55.255
        inet 192.168.55.217 netmask 0xffffff00 broadcast 192.168.55.255
        inet 192.168.55.218 netmask 0xffffff00 broadcast 192.168.55.255
        inet 192.168.55.219 netmask 0xffffff00 broadcast 192.168.55.255
        inet 192.168.55.220 netmask 0xffffff00 broadcast 192.168.55.255
        inet 192.168.55.221 netmask 0xffffff00 broadcast 192.168.55.255
        inet 192.168.55.222 netmask 0xffffff00 broadcast 192.168.55.255
```

The laptop initially takes quite a load hit on starting the smbd/nmbd, named and dhcp processes, but after approximately 60 seconds the system is pretty lightly loaded. The nmbd processes wake up periodically to broadcast the NetBios machine names to the world over our 802.11 link.

```
load averages:  0.61,  0.71,  0.48    12:53:06
63 processes:  1 running, 62 idle
CPU states:  0.5% user,  0.0% nice,  0.6% system,  0.0% interrupt,  98.9% idle
Memory: Real: 38M/55M act/tot  Free: 18M  Swap: 0K/105M used/tot

  PID USERNAME PRI NICE  SIZE   RES STATE WAIT     TIME    CPU COMMAND
18743 root       2    0  324K  916K idle  select   0:02  0.00% sshd
 6238 root       2    0  868K  944K sleep select   0:00  0.00% nmbd
31133 root       2    0  912K  956K sleep select   0:00  0.00% nmbd
 1342 root       2    0  852K  944K sleep select   0:00  0.00% nmbd
19384 root       2    0  860K  944K sleep select   0:00  0.00% nmbd
 8373 root       2    0  888K  992K sleep select   0:00  0.00% nmbd
11171 root       2    0  856K  944K sleep select   0:00  0.00% nmbd
 2741 named      2    0 1876K 2112K idle  select   0:00  0.00% named
17388 root       2    0  908K  964K sleep select   0:00  0.00% nmbd
 6453 root       2    0  884K  936K sleep select   0:00  0.00% nmbd
 9520 root       2    0  880K  928K sleep select   0:00  0.00% nmbd
24018 root       2    0  908K  940K sleep select   0:00  0.00% nmbd
16936 root       2    0  860K  944K sleep select   0:00  0.00% nmbd
 3892 root       2    0  912K  940K sleep select   0:00  0.00% nmbd
20501 root       2    0  900K  944K sleep select   0:00  0.00% nmbd
 8686 root       2    0  900K  944K sleep select   0:00  0.00% nmbd
18383 root       2    0  892K  944K sleep select   0:00  0.00% nmbd
```

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

Here is a snapshot of our new 802.11 AP traffic from the second OpenBSD laptop, running dstumbler.[21]

```
> [ 6] Marketing  (0c:00:0f:0f:0e:0e)  bn161:255:094 a SSID: Marketing
                                                     r BSSID: 0c:00:0f:0f:0e:0e
                                                     o Mfg: N/A
                                                     i Channel: 6    11.0/100
                                                     c Signal/Noise: 161/255/94
                                                     m First Seen: 18:19:15
                                                     x Last Seen:  18:19:38
-------------------------------------------------------------------------------
219:255:036 -------+++++++++++++++++++++++++++++++++ x [ basic navigation ]----
202:255:053 ----------+++++++++++++++++++++++++++++++ x [+/-]: ap up/down
199:255:056 ----------+++++++++++++++++++++++++++++++ x [</>]: node up/down
223:255:032 ------+++++++++++++++++++++++++++++++++++ x [u/d]: page ap up/down
214:255:041 --------+++++++++++++++++++++++++++++++++ x [e/h]: end/home
219:255:036 -------+++++++++++++++++++++++++++++++++ x [n/s]: newest/sort
225:255:030 ------+++++++++++++++++++++++++++++++++++ x [a/r]: autosel/resolve
217:255:038 --------+++++++++++++++++++++++++++++++++ x [o/i]: nodes/audio
228:255:027 -----+++++++++++++++++++++++++++++++++++ x [m/k]: menu/refresh
204:255:051 ----------+++++++++++++++++++++++++++++++ x [c/.]: chanlock/comment
228:255:027 -----+++++++++++++++++++++++++++++++++++ x
221:255:034 -------+++++++++++++++++++++++++++++++++ x [ file commands ]-------
213:255:042 --------+++++++++++++++++++++++++++++++++ x [l/b]: load/backup
202:255:053 ----------+++++++++++++++++++++++++++++++ x [q]:   quit
161:255:094 ----------------------+++++++++++++++++++
-------------------------[ dstumbler v1.0 by h1kari - (c) Dachb0den Labs 2001 ]
```

This shows a client associated with our OpenBSD 802.11 Access Point.

```
wicontrol -l

1 station:
00:30:65:1f:e7:f8  asid=03a0, flags=3<AUTH,ASSOC>, caps=1<ESS>,
rates=f<1M,2M,5.5M,11M>, sig=42/0
```

This is from a prismdump[22] running on the second OpenBSD machine when someone scans our OpenBSD access point.

```
- [ff:ff:ff:ff:ff:ff <- 0:2:2d:5f:19:15 <- ff:ff:ff:ff:ff:ff]
- port: 7 ts: 4048.683158 96:255 20:0
- sn: 28976 (51:55:f4:15:5d:51) len: 17
  - ** mgmt-probereq **
    - ssid: [Marketing]
    - rates: 1.0 2.0 5.5 11.0

- [ff:ff:ff:ff:ff:ff <- 0:2:2d:5f:19:15 <- ff:ff:ff:ff:ff:ff]
- port: 7 ts: 4048.683158 96:255 20:0
- sn: 28976 (51:55:f4:15:5d:51) len: 17
  - ** mgmt-probereq **
    - ssid: [Marketing]
    - rates: 1.0 2.0 5.5 11.0
```

[21] dstumbler is part of the bsd-airtools utilities http://www.dachb0den.com/projects/dstumbler.html

[22] prismdump is part of the bsd-airtools utilities http://www.dachb0den.com/projects/dstumbler.html

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

This partial log shows what a BSD or Linux “drive by laptop” user would see when running nbtscan against our IP address space.

```
Doing NBT name scan for addresses from 192.168.55.0/24

NetBIOS Name Table for Host 192.168.55.1:

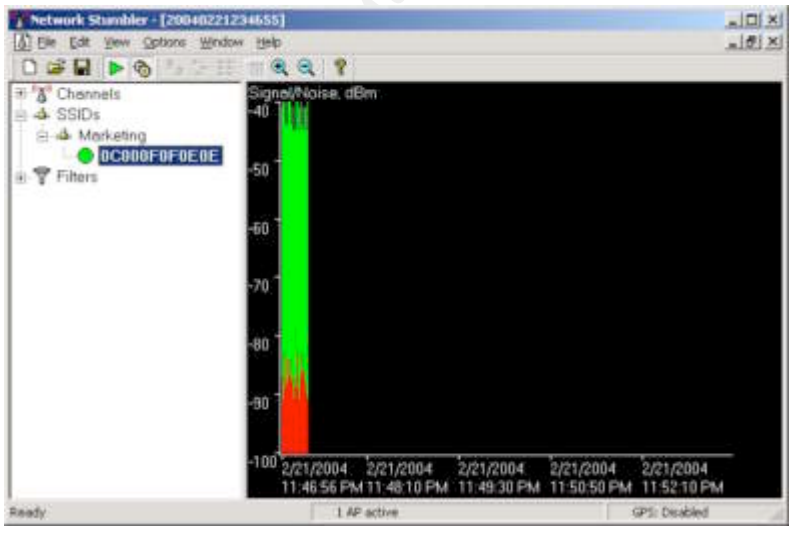
Incomplete packet, 227 bytes long.
Name             Service          Type
----------------------------------------
MARKETINGSERVER Workstation Service
MARKETINGSERVER Messenger Service
MARKETINGSERVER File Server Service
__MSBROWSE__ Messenger Service
MARKETING       Workstation Service
MARKETING       Master Browser
MARKETING       Unknown service (code 1e)

Adapter address: 00-00-00-00-00-00
----------------------------------------

NetBIOS Name Table for Host 192.168.55.202:

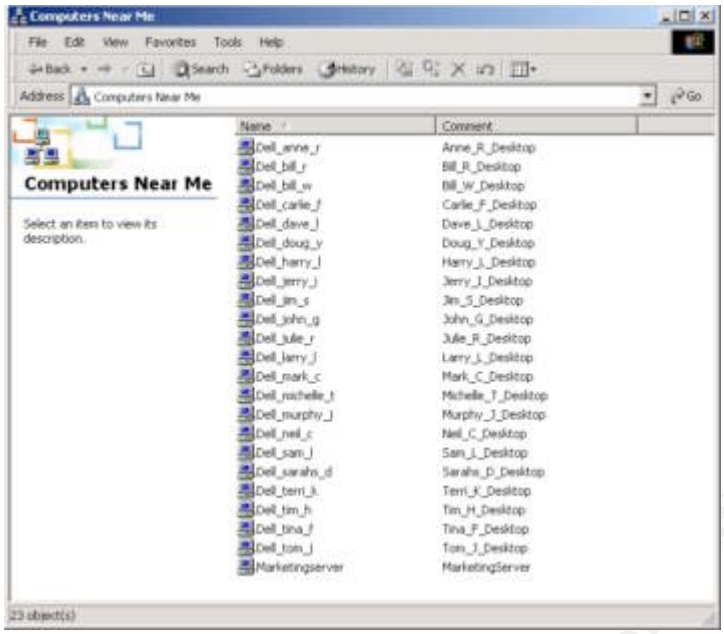
Incomplete packet, 191 bytes long.
Name             Service          Type
----------------------------------------
DELL_BILL_R     Workstation Service
DELL_BILL_R     Messenger Service
DELL_BILL_R     File Server Service
MARKETING       Workstation Service
MARKETING       Unknown service (code 1e)
```

This screen shows what a Windows “drive by” user will see when scanning the OpenBSD Access point using the Windows application NetStumbler.



© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

This screen shows what a Windows “drive by” user will see when connecting to the OpenBSD Access point when browsing “Network Neighborhood”.



VI. Conclusion

I have operated the honeypot system for several weeks, at home, at work and occasionally from my vehicle. My home is on a slight hill, with a large open area overlooking houses on two sides. Near my home there is a major thoroughfare within 802.11 broadcast distance, especially when the WarDriver is using an external antenna. I detected my first scan within 30 minutes, likely by a traveling “WarDriver” running an 802.11 access point mapping tool. I encountered my first SMB file access within 72 hours, where the attacker retrieved several attractively named documents on the “Marketing file server” with names such as FiveYearPlan.doc, EmployeeSSNum.xls, and Q4FY04_financials.xls. In actuality, the information contained within these files were typically classic RFC’s and CERT bulletins such as:

> RFC 3514, The Security Flag in the IPv4 Header; ie; the “evil bit”
> “Firewalls [CBR03], packet filters, intrusion detection systems, and the like often have difficulty distinguishing between packets that have malicious intent and those that are merely unusual. The problem is that making such determinations is hard. To solve this problem, we define a security flag, known as the "evil" bit, in the IPv4 [RFC791] header. Benign packets have this bit set to 0; those that are used for an attack will have the bit set to 1.”[23]

[23] RFC 3514 Security flag in the Ipv4 header http://www.ietf.org/rfc/rfc3514.txt

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

> "CERT Advisory CA-96.13 – "Independence Day" Alien/OS Vulnerability
> The vulnerability allows the insertion of executable code with root access to key security features of the operating system. In particular, such code can disable the NiftyGreenShield (tm) subsystem, allowing child processes to be terminated by unauthorized users."[24]

I imagine the person that accessed these documents on my machine may have been disappointed that they did not contain the data the filenames suggested. Was this person a bored 15 year old? Could it be neighbors unsuccessfully trying to access their own 802.11 AP? Or could it be competitors or criminals seeking proprietary information? The point is, we really don't know, given the anonymous nature of 802.11 links.

In my testing, I have detected numerous wireless reconnaissance efforts directed at my 802.11 AP honeypot system at work. I am preparing a presentation for my company's management listing the data I have acquired so far. By demonstrating actual 802.11 AP scanning and attempted access to the 802.11 honeypot network systems at our physical location, I can prove that the risk of deploying 802.11 systems is real, and not a theoretical abstraction.

Laptops configured in this manner are not only useful as honeypot systems, but can be used as an inexpensive solution for ongoing monitoring for 802.11 probe efforts. OpenBSD laptops fitted with wireless cards and tools such as kismet and bsd-airtools can be effectively used for auditing to discover rogue access points set up in violation of corporate wireless policies.

VII. Futures

This system is a work in progress. I will post additional information and data obtained related to this project on the website at http://www.2112systems.com/

I will also post my build scripts to generate the various configuration files, dedicated code I am experimenting with to replace the client machine smbd / nmbd processes, as well as logging and analysis tools.

I am working on hardening and sandboxing this system using OpenBSD's "systrace". This is ongoing the results will likewise be posted.

[24] Forged CERT bulletin "AlienOS Independence" http://www.xse.com/leres/gems/alienos.html

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

There are newer Prism 2.5 and Prism 3, hostAP compatible 802.11 cards that have higher RF output. These cards would be more appropriate if new cards were to be purchased.

The NL-2511CD PLUS
http://www.netgate.com/EL2511.html

The SMC SMC2532W 200mw card looks interesting
http://www.smc.com/index.cfm?action=products_show_product&productcode=SMC2532W-B

Other resources for constructing self contained hostAP based system include:

The Seattle Wireless hardware comparison page
http://www.seattlewireless.net/index.cgi/HardwareComparison

Absolute Value Systems
http://www.linux-wlan.com/products.html

VIII. References

Wearden, Graeme. "Senior management 'slow to understand wireless risks'". ZDNet UK. November 20, 2003
http://news.zdnet.co.uk/communications/wireless/0,39020348,39118028,00.htm

Leyden, John. "Wi-Fi hacker caught downloading child porn." The Register. Nov 24 2003. http://www.securityfocus.com/news/7514

Poulsen, Kevin. "Wireless hacking bust in Michigan." SecurityFocus Nov 12 2003. http://www.securityfocus.com/news/7438

Luria, Mary M. and Kibel, Gary A. "Corporate Liability For Online File Sharing" Davis and Gilbert LLP http://articles.corporate.findlaw.com/articles/file/01009/009389

Airsnarf - A rogue AP setup utility 0.2. The Shmoo Group http://airsnarf.shmoo.com/

Brandt, Andrew. "A latte, a Wi-Fi link and a hacker" PC World Nov 5, 2003
http://www.computerworld.com/mobiletopics/mobile/story/0,10801,87523,00.html?f=x68

Ezor, Jonathan I. "Free Wi-Fi Hotspots and Potential Legal Problems" BizLawTech - The Official Web site for the Institute for Business, Law and Technology at Touro Law Center Jun 2, 2003 http://iblt.tourolaw.edu/blog/archives/000010.html

NetStumbler.com "The world of WiFi" http://www.netstumbler.com/

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

Kismet is an 802.11 layer2 wireless network detector, auditing tool and sniffer http://www.kismetwireless.net/

bsd-airtools is a package that provides a complete toolset for wireless 802.11b auditing, including dstumbler and prismdump by dachb0den labs http://www.dachb0den.com/projects/bsd-airtools.html

Nmap ("Network Mapper") is an open source utility for network exploration or security auditing http://www.insecure.org/nmap/

Honeypots – “Definitions and Value of Honeypots” Lance Spitzner http://www.tracking-hackers.com/papers/honeypots.html

The OpenBSD FAQ, which covers obtaining and installing OpenBSD is located at http://www.openbsd.org/faq/index.html

The up to date list of current OpenBSD patches, as well as the patches themselves, which are supplied as source code, is available at http://www.openbsd.org/errata.html

The dmassage utility is available at http://www.sentia.org/projects/dmassage/

The Intersil Prism 802.11 card update utility is available at ftp://ftp.dlink.com/Wireless/DWL650/Firmware/

I found Prism 2 card 1.4.9 firmware at http://linux.junsun.net/intersil-prism/firmware/1.4.9/

Jun Sun's Mini-howto on Flashing Intersil Prism Chipsets is at http://linux.junsun.net/intersil-prism/

RFC 3514, The Security Flag in the Ipv4 Header; “the evil bit” http://www.ietf.org/rfc/rfc3514.txt

© SANS Institute 2004, Author retains full rights.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.